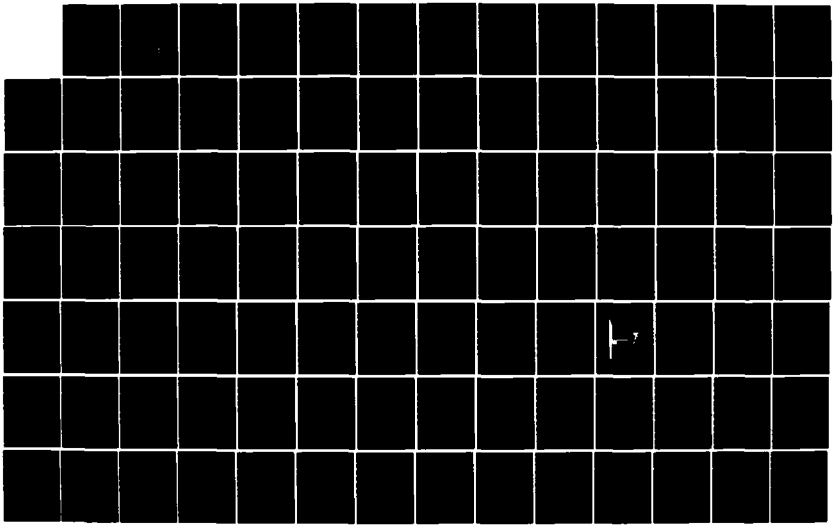
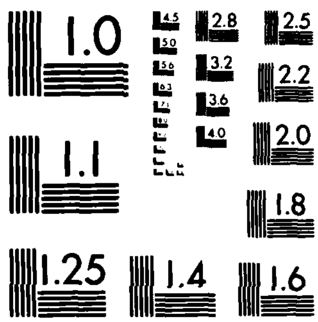
AD-A151 872 ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING B M WAECHTER DEC 84 AFIT/GOR/OS/84D-15 1/3

UNCLASSIFIED F/G 22/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFIT/GOR/OS/84D-15

ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT

THESIS

Brian M. Waechter
Captain, USAF

AFIT/GOR/OS/84D-15

Approved for public release; distribution unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GOR/OS/84D-15	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering AF Institute of Technology 6b. OFFICE SYMBOL (If applicable) AFIT/ENS	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See box 19	

12. PERSONAL AUTHOR(S)
Brian M. Waechter, Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1984 December	267

16. SUPPLEMENTARY NOTATION

from backup

17. COSATI CODES		
FIELD	GROUP	SUB. GR.
22	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Spacecraft debris, computerized simulation, space stations, Computer Programs, Fortran, Subroutines, Theses.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT

Thesis Advisor: Mark M. Mekaru, Lt Col, USAF
Asst Professor of Operations Research

Approved for public release: IAW AFR 190-17.
Lynn E. Wolaver 2 Feb 85
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Mark M. Mekaru, Lt Col, USAF	513-255-5758	AFIT/ENS

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

Analysis of Space Station operations in the space debris environment involved the conceptualization and development of a simulation model to provide initial estimates concerning Space Station survivability and fuel requirements. An initial review of recent literature provided the baseline upon which to conceptualize the debris environment system, and indicated the relative insensitivity of satellite-of-interest collision probability calculations to modeling debris density with varying complexity. In addition, the literature identified that the debris population unable to be detected by current means, the rate of unintentional explosions and inter-object collisions, and the dynamics of these occurrences are important system parameters on which little is known. System conceptualization consisted of studying the relationships between those system elements identified in the literature. The predominance of unstable relationships within the system pointed toward the possibility of uncontrolled, self-sustaining growth in the space debris population. Conceptual model elements significantly affecting the space debris population and lending themselves to modeling were included in the discrete-event SLAM simulation model developed. The model simulated space debris environment dynamics up to Space Station initial operations, through its growth period, and beyond Space Station system maturity. Model results indicated that at least one collision could occur within the first 29 years of Space Station operations. The results involving the number of encounters with debris requiring that the Space Station perform avoidance maneuvers showed that the spacecraft's fuel capacity will be insufficient should the Space Station be required to maneuver away from an object any further away than one kilometer. The~~se~~ results stress the need for greater consideration of the survivability of large, long-term spacecraft in such an environment, and for greater ground-tracking or on-board debris detection capabilities.

Originator-supplied Keywords included: → /ntp.

AFIT/GOR/OS/84D-15

Brian M. Waechter, B.S.
Captain, USAF

December 1984

ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Preface

Man has approached outer space in much the same way he has approached other new resources placed at his disposal by assuming it is limitless. However, man is beginning to learn that even space is becoming a limited resource. The quantity of man-made debris orbiting the earth is quickly becoming a very real hazard to satellites on which man is increasingly dependent. While research continues on the space debris environment, very little work has considered the potentially greater risk posed to large, manned structures such as the proposed Space Station.

I chose this subject because I felt that the international and corporate interest in the Space Station as well as the scientific importance and sizeable financial investment in the project warranted a specific look at its long-term survivability in the space debris environment. If my research results should paint a gloomy picture, action would have to be taken soon to prevent an unfortunate mishap at some future time.

While only my name appears on the title page, other individuals played integral roles in helping me with this research effort. I am forever indebted to my faculty advisor, Lt Col Mark M. Mekaru, who not only

supported and guided me, but even more importantly, taught me about leadership, officership, and professionalism. I also wish to thank Dr. William Wiesel for his assistance during the conceptualization stage of the thesis. A word of thanks goes to Mrs. Londa Wilkes, who transformed my atrocious handwriting into a work of art. Finally, I wish to thank my family and close friends for their understanding on those numerous occasions when my research took time away from them.

Brian M. Waechter

Table of Contents

List of Figures

Figure		Page

List of Tables

Abstract

Analysis of Space Station operations in the space debris environment involved the conceptualization and development of a simulation model to provide initial estimates concerning Space Station survivability and fuel requirements. An initial review of recent literature provided the baseline upon which to conceptualize the debris environment system, and indicated the relative insensitivity of satellite-of-interest collision probability calculations to modeling debris density with varying complexity. In addition, the literature identified that the debris population unable to be detected by current means, the rate of unintentional explosions and inter-object collisions, and the dynamics of these occurrences are important system parameters on which little is known. System conceptualization consisted of studying the relationships between those system elements identified in the literature. The predominance of unstable relationships within the system pointed toward the possibility of uncontrolled, self-sustaining growth in the space debris population. Conceptual model elements significantly affecting the space debris population and lending themselves to modeling were included in the discrete-event SLAM simulation model developed. The

model simulated space debris environment dynamics up to Space Station initial operations, through its growth period, and beyond Space Station system maturity. Model results indicated that at least one collision could occur within the first 29 years of Space Station operations. The results involving the number of encounters with debris requiring that the Space Station perform avoidance maneuvers showed that the spacecraft's fuel capacity will be insufficient should the Space Station be required to maneuver away from an object any further away than one kilometer. These results stress the need for greater consideration of the survivability of large, long-term spacecraft in such an environment, and for greater ground-tracking or on-board debris detection capabilities.

I. Introduction

Background

> "... current practices, which leaves debris in orbit, could lead to a state where risk of collision for operating spacecraft may not just become significant, but might even preclude using certain regions of space in the future due to an uncontrolled growth in the amount of debris." (31:9)

> "... continuation of present policies and practices ensures that the probability of collision will eventually reach unacceptable levels, perhaps within a decade." (33)

These start warning have brought increased attention and concern from the scientific, governmental, and legal communities over the one environment that seemed endless and inexhaustible--space. Past practices by all nations utilizing space since 1958 have created a cluttered environment where satellites must operate with increasing risk. Ironically, the realization of the increasing momentum of this problem comes at a time when many nations are becoming increasingly dependent on their space activities for communications, military, research, and future production needs.

Many scientists who studied the problem share the same predictions as presented in the above quotations,

and several unexplained recent events in space point to the increasing collision risk. kessler, Landry, Cour-Palais, and Taylor propose that a U.S. land-use satellite, Geos 2, a U.S. Pageos balloon, and the Soviet satellite Cosmos 954, which achieved notoriety for its uncontrolled reentry over northern Canada in 1978, all reached their fate because of collisions with orbiting space debris. These scientists go on to predict a major collision within the next six years if the debris population continues its present rate of growth, and a total of three collisions by 1995 (18:37).

While scientists do not all agree as to what this "rate of growth" is or will be, all agree that it is large and is increasing. Kessler, Cour-Palais, and Perek agree that the debris density is increasing exponentially with time (25:115; 14:2645). Kessler contends that this exponential increase will occur even if no more objects are put into the environment than the number that reenter the atmosphere (14:2645). This argument is made with the belief that once the debris density reaches a certain point, inter-object collisons will become the dominant source of orbital debris, completely out of human control (15). Such a scenario brings to question the reliabiity, cost-effectiveness, and safety of increasingly important satellites and

present and future manned spacecraft.

While the possibility of a collision with a satellite due to the meteroid flux has been present since the first satellite launch, concern over man-made debris has recently taken precedence for two reasons. First, Kessler's prediction of uncontrolled debris population growth due to inter-object collisions would produce a flux of orbital debris exceeding the meteoroid flux (15). Second, man-made debris is a unique problem since it remains in orbit, unlike meteoroids which are transient through the various altitudes (35:2). This orbiting debris consists of operational and de-activated payloads, mission-related debris (rocket bodies, clamps, shrouds, etc.), and explosion/collision fragments (5:192;33). While it may at first appear that the smaller mission-related debris and explosion/collision fragments need not concern the space community, the combination of significant relative velocities with composite materials and delicate structures, such as solar arrays, make even the smallest particle a formidable projectile. In fact, Perek calculated that a collision at 10 km/sec will eject 115 times the mass of the impacting debris from the satellite (25).

As the lethality and increasing numbers of man-made debris have become known, recommendations have been

made concerning the control of the debris environment and survivability enhancements in future spacecraft. Proposals for debris control include non-operational satellite retrieval, placement of inactive satellites into a designated "junk" orbit, controlled reentry of inactive satellites, and design requirements that reduce the release of non-functional and/or explodable objects from a launch (18:37, 33; 15; 9:365). Survivability enhancements include on-board collision detection and avoidance systems, shielding, and bumpers (30: 33). The problem all of these recommendations share is that most of them are all very expensive, some even being economically impossible at the present time, and as far as the debris control recommendations are concerned, require international coordination and cooperation. The point at which a proposal becomes economically feasible and international cooperation becomes realizable depends on both past events and thorough analysis and prediction of future events. Only with a thorough understanding of the space debris environment can we make well-informed decisions about how we should address the problem based on its magnitude.

There has not been a lack of analysis on the satellite collision hazard problem. However, many studies have come up with contradictory results as to

the criticality of the situation, and almost all have considered only smaller satellites, including the Space Shuttle, when calculating collision probabilities. Indeed, scientists such as Perek, Kessler, Reynolds, Fisher, and Rice concur that the acceptable level of risk will decrease with time, higher altitudes than the Shuttle, and larger structures (25; 15;30:285). One such "larger structure" that has received little attention with regards to debris hazard analysis is the proposed Space Station. The Space Station Program Description Document prepared by the Space Station Task Force acknowledges the collision hazard only twice. First, the document discusses the impact resistance of the spacecraft, but only in terms of the meteoroid flux (22: Sec 6, 3). Second, the document expresses concern over the interaction of composite materials with both man-made and meteoroid debris particles. It goes on to state:

> "No data are available to assess the debris impact threat to large space systems or to evolve designs to minimize this threat. Likewise, there has not been an evaluation of these materials for possible fragmentation and debris generation." (23:Sec 5, 18)

The establishment of a permanent manned presence in space puts increased importance on debris hazard work

involving the Space Station for several reasons. First, the Space Station will be many times larger than any other manned spacecraft previously put into space. Therefore, it is much more likely of being hit by debris. Second, this larger target will be manned, therefore increasing concern over system survivability. Third, the Space Station will be permanent. Therefore, it will not only be a more susceptible target because of its size, but will also be more likely to be hit by debris because it will constantly be exposed to such an environment. Fourth, it is safe to say that the Space Station will be the single most concentrated effort and probably the most expensive effort since Apollo, so great care will be taken to ensure the program's success. Finally, the Space Station will be open to international and commercial use. Overall, a system, seemingly very susceptible to collisions with space debris yet housing human beings whose safety is paramount, will exist in a situation where many countries and commerical firms have a very real stake in its success. It is apparent that all of the elements exist to accelerate the progress in space debris control, should collision analysis involving the Space Station yield significant results.

Purpose

The purpose behind my research is, through simulation, to place the Space Station in a dynamic man-made space debris environment to calculate collision probabilities from system initial operational capability (IOC), through its enhancement, and into the period of system maturity. I will also obtain the number of "close calls" over intervals of time which, with advance notice, would require that the Space Station be repositioned to avoid a collision. I will use these values to analyze the extent to which this repositioning frequency affects Space Station fuel usage and resupply requirements. System simulation allows for both the analysis of individual entities within the system and of interrelationships between entities. Since the space debris environment is characterized by many remaining unknowns, simulation provides the excellent medium with which to apply sensitivity analysis to assumptions and theoretical parameters.

Methodology

Since numerous studies exist which analyze the collision hazard for systems other than the Space Station, my overall methodology is to tie the information from these studies together and extend it to analyzing the Space Station. This is not an easy

task, since each study involves an environment where many parameters remain unknown. One such critical unknown parameter is the actual debris population. The research will attempt to find the common denominators from the past work and incorporate them into my simulation model in hopes that the model adequately represents the cumulative findings of those experts who have worked years in this area.

The finding of these common denominators takes place during the first step of my research, which is to conduct a literature review to update the information regarding the various parameters I will use in the simulation model. The amount of information obtained will determine what elements of the model can indeed be credibly implemented and what assumptions and simplifications must be maintained.

The second step is to apply the system science paradigm to the problem. The system science paradigm consists of system conceptualization, analysis and measurement, and parametric modeling and testing. The literature review will aid in conceptualizing the entire space debris environment system, which in turn gives an understanding of all element interactions within the system. Analysis and measurement consists of defining analysis objectives, establishing a research design, and collecting data needed for the

model. Data collection will be accomplished during the literature review. Parametric modeling consists of building a simulation model using the SLAM simulation language. Testing of the model can be broken down into verification and validation, which I treat as separate steps and discuss below.

The third step is to verify the SLAM simulation model by running traces on the model parameters. The traces list the values of these parameters as the simulation run progresses in time, and analysis of the trace output determines whether discrete events are scheduled properly and occur at the appropriate time, whether variables are being assigned reasonable values, and whether these variables are effectively being passed between the necessary subroutines and simulation modules. In general, the above verification procedure should increase confidence in the ability of the parametric simulation model to accurately represent the conceptual model developed earlier.

The next step, model validation, may be difficult since very little work has been done with regard to Space Station collision probabilities previously, therefore providing few points of comparison. The literature review may provide information with which to compare certain aspects of the model such as satellite population growth over time and explosion, ASAT test,

and launch rates. Through these individual comparisons, a higher confidence level may be reached regarding the model's accuracy in simulating the actual space debris environment.

The fifth step is to collect the necessary output from the model to apply the previously established experimental design. I can then proceed with my analysis of the sensitivity of Space Station collision probabilities to varying model parameters. Assessment of the significance of repositioning fuel requirements due to possible collisions with debris consists of first obtaining data on Space Station fuel capacity, normal fuel usage, and refueling rates. Then, based upon the number of times collision avoidance is required, additional fuel usage can be calculated. Finally, I will discuss the impact of these collision avoidance maneuvers on Space Station operations.

II. Literature Review

Introduction

A review of the literature involving the survivability of spacecraft in the space debris environment provided the necessary information with which to conceptualize, develop, and analyze a system simulation model. Consensus among the experts working in this area concerning certain aspects of the environment and its analysis established a framework for the development of the model and subsequent analysis. Aspects of the environment where little or no information could be found or where common assumptions were held by previous researchers provided justification for maintaining those assumptions in my research effort. Overall, I felt a review of the literature made available to me would promote the development of a model representative of the space debris environment as is known or can be predicted.

The following sections attempt to provide a complete picture of the research conducted to date on the debris environment and its analysis. First, I will describe the way in which past studies modeled the space debris environment. I will then focus on several aspects of the environment where additional work has been concentrated. These areas include the accuracy in measuring the debris densities in space, the importance

of natural debris (meteorities, etc.) in calculating collision probabilities, and the identification of the sources of man-made space debris. Finally, I will analyze the methods by which previous researchers calculated the probability of collision between a spacecraft and debris.

Past Models

Past efforts in modeling the space debris environment can be roughly divided into two categories. The first category consists of those models where space debris density is calculated as a function of two dimensions. One is usually altitude and the other is latitude, longitude, or orbital inclination. The second category consists of those models representing space debris density as a function of altitude only, based upon results of other research efforts. The following paragraphs more fully describe the details of these models.

Donald Kessler and Burton Cour-Palais developed the first model found in the literature to treat debris densities as a function of two dimensions. They modeled the spatial density of this debris as a function of altitude and geocentric latitude. The use of these two dimensions resulted in a "grid" surrounding the earth, with a "grid cell" being three degrees in latitude and 50 kilometers in altitude

(14:2637). The existence of orbital perturbations resulting from natural phenomena led the authors to assume that the spatial density within each cell was uniformly distributed (14:2637). Kessler and Cour-Palais found the spatial density in each "grid cell" by calculating the probability of finding each debris object in a particular cell and then summing these probabilities. The spatial density for a particular cell was simply the sum of these summed probabilities for every debris object having some positive probability of being in that cell divided by the cell volume (14:2637). Robert Reynolds, Norman Fischer, and Eric Rice also used this dimensional grid method for their modeling of the debris environment and Space Shuttle hazard analysis.

Robert Reynolds, along with Norman Fischer and Eric Rice, developed a model very similar to Kessler's model, describing space debris density as a function of altitude and latitude. The Reynolds, Fischer, and Rice model followed Kessler's treatment of the amount of time a particular piece of debris spent in a particular cell by using the debris object's apogee and perigee altitudes along with its orbital inclination to determine its contribution to a particular cell object density (30:281). The authors separated the varying debris densities using an earth-centered two-

dimensional grid with altitude divided into 50 kilometer increments from 150 to 4000 kilometers and latitude spaced every five degrees (30:281). Reynolds, Fischer, and Rice felt the inclusion of latitude dependence into the model was important because the model did not then average out the debris density distribution and could take into account varying velocity distributions for the collision probability calculations (30:282). The inclusion of a debris population velocity distribution function allowed collision hazard levels to be calculated for specific orbital planes (30:280). While the debris environment models discussed thus far were quite similar in their approach, a model developed by V.A. Chobotov departed significantly in its method of collision hazard analysis.

The most significant dissimilarity between the Chobotov model and the previously discussed models is that Chobotov considered the distribution of the tracked population as a function of altitude and orbital inclination, not latitude (2:484). Chobotov did not divide the space environment into a spherical grid as did the other researchers, but instead used a particular satellite-of-interest's (SOI) orbital parameters and the concept of an "encounter sphere" to determine the amount of debris encountered (2:484).

This "encounter sphere" surrounded the satellite-of-interest and determined the debris density along the satellite of interest's flight path depending on the quantity of debris entering that sphere. The Chobotov model did have three things in common with the other models, however. First, Chobotov assumed that the positions of debris within the "encounter sphere" were randomly distributed. Second, Chobotov only considered trackable objects (those objects detected and catalogued) in his debris density compilations, as did all of the other models discussed thus far except for Kessler's model (2:484). Finally, Chobotov and the previously discussed models represented debris densities in two dimensions. However, other researchers have simplified the development of their own models by considering debris density as a function of altitude only.

The simplification of space debris environment models from considering debris densities in two dimensions to one dimension initially appears to be a step backward in accurately describing the debris environment. However, results from the two-dimensional models themselves indicate that accuracy is not lost in using only altitude to describe debris densities. After analyzing the results from his two-dimensional model, Donald Kessler stated as early as 1980 that:

> "The probability of a particular spacecraft colliding with any of these 4,719 orbiting objects is a function of that spacecraft's orbital position and velocity. However, for most types of orbits, the probability is mainly (within a factor of 2) a function of spacecraft altitude--the major exception being for spacecraft in orbits of inclinations between 100 degrees and 130 degrees, where the probability can be several times the average for that altitude" (15).

Kessler later stated that inclination was not an important parameter in debris density determination even though initial modeling results indicated the contrary. He found that tne use of only active satellite data in the calculation of debris densities and the assumption that explosion debris moved in the same direction as their source resulted in the appearance of inclination as being important. However, subsequent modeling and research showed that altitude alone could adequately describe debris densities (16). Robert Reynolds, responsible for the development of several two-dimensional models, also came to the same conclusion. In speaking of his modeling results, Reynolds stated:

> "In the course of this work, collision probabilities from the added particles were found to depend primarily on the percentage of time which the added

particles spent at the Shuttle altitude, the effect of varying inclination in the debris orbit not having as great an effect" (29:106).

Perhaps because of the relative newness of the Kessler and Reynolds findings or because of reservations concerning these findings possibly held by other researchers, I found only one model in the literature describing debris densities as a function of altitude alone. The following paragraphs describe the simulation model developed by Robert Penny and Richard Jones.

The Penny and Jones effort consisted of developing a model to simulate the space debris environment for use in calculating Space Shuttle collision probabilities and assessing various alternatives for controlling the debris environment (24:4). The researchers used the Q-GERT simulation language to simulate, in a discrete event manner, the dynamics of the debris environment. In other words, Penny and Jones determined that launches into space, anti-satellite (ASAT) tests, unplanned explosions from spent boosters, collisions between debris, and debris decay into the atmosphere all contributed to either increasing or decreasing the debris population, so these occurrences were scheduled so as to adjust the population over time (24:5). The researchers divided the debris environment into three concentric "shells"

or "bands" surrounding the earth which were based solely on altitude. They assumed that within each altitude band debris was distributed uniformly, which is consistent with the previous studies.

While the most obvious simplification between the Penny and Jones model and past models is the difference in the number of dimensions used to describe the debris densities, other simplifications in the Penny and Jones model do exist. First, Penny and Jones did not analyze the orbital parameters of each catalogued debris object to determine the exact debris densities at any instant in time. Instead, they used the debris populations averaged over time and listed by altitude in the North American Air Defense Command (NORAD) CLASSY catalog (24:33). The rationale behind this generalization was probably that a dynamic simulation model could not predict the orbital parameters of future debris objects with any accuracy, so the researchers dropped this individual accounting of debris. The same rationale could stand for another simplification--debris dwell time in a particular altitude band. Penny and Jones took dwell time into account only as much as the NORAD CLASSY data does. The third simplification involves the relative velocity between the satellite-of-interest and the debris object with which it will collide. Penny and Jones determined

that the relative velocity could take on any value from zero to twice the satellite-of-interest's orbital velocity, so they calculated the circular orbital velocities for a debris object in the middle of each of the three altitude bands and used those values as average relative velocities (24:32-33). This is quite different from the previous models, which used velocity distribution functions of varying complexity based upon velocity calculations for each catalogued object.

Although there are several significant simplifications present in the Penny and Jones model, the model does possess several advantages over past models. First, the model is dynamic in nature, so sensitivity analysis on the many remaining unknowns concerning the space debris environment can be performed with relative ease. Second, this model is the only model other than the Kessler model that includes space debris too small to be detected by NORAD in the debris environment. As we will see later in this chapter, this untrackable debris is probably the most critical parameter in the collision hazard problem for which we have almost no concrete data.

In addition to using the above information as a baseline for the development of my model, I decided to study certain aspects of the debris environment in more detail in an attempt to learn what the limitations,

assumptions, and inputs into my model would be. These elements of the space debris environment, as I will call them, consist of the tracked and untracked man-made debris populations, and the sources of debris which include launches, inter-object collisions, ASAT tests, and unintentional explosions.

Man-made and Natural Debris Populations

As stated in chapter one, the man-made space object population can be classified as functional payloads or as non-functional payloads and debris. Of those objects which can be tracked by radar or optical means, 95% falls into the second category (15). On top of the fact that there is so much litter in space is that most of it can be found at altitudes between 500 and 1000 kilometers, with a maximum density near 850 kilometers (35:3). This fact obviously prompted the research discussed earlier to concentrate on the collision hazard presented to the Space Shuttle which operates at low altitudes, and remains applicable to my analysis concerning the planned Space Station. Despite the ability to track a portion of the debris in space, uncertainties about exact size and orbital parameters make the collision hazard problem more complex for even these objects on which we have some information. The problem becomes several orders of magnitude greater when we consider debris which is too small to be

detected, and therefore of which we have no information.

The untracked man-made debris population consists of those objects which cannot be detected by the radar and optical means available to NORAD. Detection capability varies with altitude, but, in general, the untrackable population is made up of objects less than 10 centimeters in diameter at 1000 kilometers altitude or objects less than 4 centimeters at 500 kilometers altitude (15; 35:15). This class of extremely small objects in space would not be of any concern if (1) it did not present a hazard to spacecraft because of the small size of the debris and if (2) the number of these objects was insignificant with respect to the overall debris population even if the objects could produce considerable damage. Unfortunately, the untrackable population is of great concern exactly because of its lethality despite its size and its predicted large numbers.

The relatively small size of undetectable debris in space appears at first as being unable to cause significant damage to a spacecraft, but the extremely high velocities at which these objects travel, on the order of kilometers per second, makes the exact opposite conclusion true. Ballistic research verifies the severity of a collision with these small objects.

Particles as small as one millimeter in diameter can cause structural damage to a spacecraft with a single impact (35:12). Furthermore, particles of .001 millimeters or less which are found in the exhaust of conventional solid rocket motors can cause a "sand blasting effect" on sensitive optical surfaces (35:11). Although estimates concerning the size of the untrackable population vary, researchers agree that the number of objects is quite large. Donald Kessler believes that the size of the untracked population increases over the tracked population with increasing altitude, which follows from the limitations of tracking devices at higher altitudes (15). Thus, Kessler believes:

> "... a sufficient reservoir of small, untrackable objects at a higher altitude must exist to produce a continuous flow of objects 'raining down' through lower altitudes due to atmospheric drag" (15).

A NORAD test in 1978 calculated the unobserved debris population to only be between 7 to 14% of the observed population, but other researchers believe it could currently be anywhere from one to ten times as high (15; 24:110). Recent tests using the United States Air Force GEODSS telescope system which is designed to track objects in geosynchronous orbit resulted in the identification of approximately 40,000 1-centimeter

size objects in low- and medium-Earth orbits. The number of these particles of just this one size alone is about eight times the current tracked population (16; 36:16).

Proliferation of both trackable and untrackable man-made space debris overshadows another class of debris that was of some concern before anyone realized the potential man-made debris hazard. Meteorites, which unlike man-made debris are far more transient through low earth orbit (LEO), consist of particles with diameters seldom greater than one centimeter (8:4). Even through researchers have shown that debris of even these small sizes can cause considerable damage, very few individuals interested in the debris collision problem have included natural debirs in their calculations.

There are several possible reasons for this. First, the man-made debris flux has actually surpassed the natural debris flux in recent years. Second, man-made debris has more than one chance to collide with a spacecraft since it is in LEO itself. Third, since past studies have to some extent proposed or investigated ways to control the debris environment, only that debris which can be controlled was considered. Therefore, it appears that there is a consensus that the meteoroid flux is inconsequential in

calculating collision probabilities (4; 37).

Sources of Debris

In addition to getting a feeling for what debris should be considered in constructing a simulation model, the sources of that debris must be investigated to learn of the dynamics of the debris growth rate. Sources of debris can essentially be broken down into launches, collisions between debris objects, and both intentional and unintentional explosions. The follow-wing paragraphs given an indication of the rate at which these events occur and the contribution they make to the space debris population.

Spacecraft launches not only immediately contribute to the debris flux by placing launch-related debris into orbit, but also place spent boosters into orbit which later can become sources of unintentional explosions. Launch rates have increased steadily since 1958, and currently 120 to 150 new payloads are launched per year (18:38; 16). According to Kessler, these occurrences cause the debris population to increase by approximately 11% per year (18:38). As major contributors to the immediate and future debris population, launches are also the most easily controlled source of debris.

Inter-object collisions are sources of debris which are by no means controllable because they are purely a

function of the existing debris population. Although the demise of several satellites is believed to have been caused by collisions with other objects, there is no way to accurately assess nor predict inter-object collision rates. The debris generated from these collisions is a function of the colliding objects' relative velocities and sizes, and very little work has been done in documenting the size, number, and dispersion of the resulting debris. However, it is generally believed that inter-object collisions cause the overall debris population to increase at a geometric rate (35:2).

The sources of debris which contribute the most to debris population growth are both intentional and unintentional explosions. In fact, researchers estimate that 60% of the tracked debris is a result of both unintentional and intentional explosions (18:38). Intentional explosions, primarily consisting of ASAT tests, account for 14% of the 60% trackable debris mentioned above (18:38). Nicholas Johnson, who conducted extensive research on the Soviet ASAT test program, stated that Soviet ASAT tests account for 9% of the trackable debris as of July of 1982 (13:358). This relatively low percentage could be misleading, however. As a result of earlier research in this area, Kessler concluded that:

> "The relatively small number of observed fragments generated by the eight USSR anti-satellite tests may be misleading. High intensity explosions produce a very large number of small, unobservable fragments. Thus, their contribution to the total debris picture could be much larger" (15).

The majority of debris resulting from explosions in space comes from explosions which are unintentional. Defective boosters are primarily responsible for these explosions, of which eleven have involved U.S. boosters (12:51). It is obvious that both intentional and unintentional explosions contribute substantially to the debris population, particularly to that debris which cannot be detected. Unfortunately, understanding the actual dynamics of these important sources of debris has yet to be accomplished, so uncertainties remain and assumptions must be made when attempting to model the space debris environment.

The uncertainties that exist in debris generation from launches, inter-object collisions, and explosions concern the number of debris generated, the size of the resultant debris, and the dispersion of these particles. The amount of debris produced as well as the size distribution of the debris depends very much on the type of collision or explosion, and particle dispersion has a direct impact on debris density

distribution. Therefore, these uncertainties greatly affect the ability to accurately model the occurrences of launches, collisions, and explosions. Past modeling efforts have used data from ground explosion and hypervelocity collision tests which were extrapolated to approximate actual energy levels in space (35:21). Much work has yet to be completed until the dynamics involving these sources of debris are known with certainty.

Collision Probability Derivations

A review of past literature indicates that the ways in which collision probabilities have been calculated are as varied as the models for which they have been used. Despite the differences in their approaches, researchers share a common understanding of the important parameters involved in calculating collision probabilities. These parameters consist of the spatial density of space debris, the relative velocity between the satellite-of-interest and the debris object encountered, the area of the satellite-of-interest, and the amount of time the satellite must operate in the debris environment (2:484; 25; 30:279). Past efforts have included these parameters in varying detail into collision probability calculations using integration, the Poisson probability distribution, and reliability theory. The following sections look at each of these

approaches in more detail as well as approaches which did not involve the relative velocity parameter.

Robert Reynolds, Norman Fischer, and Eric Rice used path integrals in formulating collision probability equations, then altered these equations based on the structure of their model (13:280). The derivation involved using integration to calculate debris densities and relative velocities, and using the SOI cross-sectional area averaged over all aspect angles to come up with the frequency of collision equation:

$$\dot{C}(\vec{r},t) = \sigma_{tot} \int d^3v(\vec{v}-\vec{v}_p)f(\vec{r},\vec{v},t) \qquad (30:281)$$

where

σ_{tot} = collision cross section

d^3v = volume element in velocity space

$\vec{v}$ = speed relative to the atmosphere

$\vec{v}_p$ = velocity of object through debris population

$f(\vec{r},\vec{v},t)$ = phase space number density function for the debris population (30:279-281)

The Reynolds, Fischer, and Rice model, if you recall, involved dividing the debris environment into cells of certain latitude and altitude dimensions. Because the authors depicted the SOI as moving through regions of constant debris conditions from cell to cell, they replaced the above path integral with a summation over the cells traversed of the form:

$$P_c(t_o) = \sigma_{tot} \sum_i n(i)V_r(i)t_i$$

where

i = index running over cells traversed

t_i = time spent traversing cell i

$t_o = \sum t_i$

$N(i)$ = number density of debris averagd latitude (30:282)

This transformation greatly simplified the collision probability calculation yet maintained the inclusion of the major contributing parameters.

Derivations Involving the Poisson Distribution. Prior to his work with Norman Fischer and Eric Rice, Robert Reynolds fitted his probability of collision equations involving integration to the Poisson distribution (29:107). Reynolds' method of calculation involved the distance the SOI traveled between collisions, or what he called the mean free path length (2). The author defined the mean free path length as a function of the average cross-section and number density of the scattered debris particles, written in equation form as:

$$\lambda = 1/\sigma_s n_s$$

From the Poisson distribution, the probability that the SOI moved a distance "x" without suffering a collision was:

$$P_{nc}(x) = \exp(-\sigma_s n_s x)$$

Expressing distance in terms of particle speed and

travel time:

$$P_{nc}(t) = \exp(-\sigma_s n_s v_r t)$$

where

v_r = mean local relative velocity of particles with respect to the Space Shuttle

Reynolds went on to calculate the probability of one or more collisions using the equation:

$$P_c(t) = 1-P_{nc}(t) \qquad (29:14)$$

The author simplied this calculation by observing that in LEO, most debris particles are in circular orbit. Therefore, Reynolds concluded that a particle's transverse velocity (parallel to the earth's surface) is much greater than its radial velocity (normal to the earth's surface) which enabled him to only consider transverse velocities in his calculation of relative velocities (29:119). Reynolds also observed that all of the parameters involved in $P_c(T)$ except for n_s were essentially constant. Therefore, he concluded that that probability of collision was directly proportional to and solely dependent on the debris density parameter (29:103).

Martin Hechler and Jozef Van der Ha used the same basic approach as Reynolds in deriving a collision probability equation. The authors' calculations involved collision probabilities between an SOI and other active satellites in the geostationary orbit

(9:361). Hechler and Van der Ha used a Monte-Carlo-type method involving satellite spatial densities and relative velocities to derive a nine-dimensional integral (9:362). The authors then used that equation to calculate the collision rate per year, which they defined as λ. Using the Poisson distribution, they formulated the probability of at least one collision occurring in w years as:

$$P(W<w) = 1-\exp(-\lambda w)$$

where

w = time elapsed with no collisions (9:364)

This equation is actually identical to the equation developed by Penny and Jones using reliability theory, which I will discuss in the next section.

Derivation Involving Reliability Theory. Penny and Jones used reliability theory to derive the collision probability equation used in their modeling effort. They proposed that a component's reliability, or probability of survival, as a function of time is:

$$R(t) = \exp(-\lambda t)$$

where

λ = chance or random failure rate

The probability of failure is therefore:

$$Q(t) = 1-\exp(-\lambda t) \quad (24:7)$$

Equating a component failure with a collision between the Space Shuttle and a debris object, Penny and Jones

defined λ as the collision rate, or in other words, the number of debris objects encountered per unit time. Based upon the structure of their model, this became the number of debris objects being swept out by the Space Shuttle. With the volume swept out per unit time given as Av and the Satellite density in the altitude band as ρ , Penny and Jones calculated the collision rate as:

$$\lambda = \rho A v \Delta t$$

The final form of the collision probability equation thus became:

$$P(\text{collision}) = 1-\exp(-\rho A v \Delta t) \qquad (24:8)$$

It is apparent that this formulation is identical in form to those developed directly from the Poisson distribution, and the parameters involved are very close to those discussed earlier. However, while Reynolds concluded that the probability of collision depended only on the number of debris objects, Penny and Jones assumed debris density, relative velocity, and the calculation interval to be constant and collision area as being the only parameter to vary from calculation to calculation (24:10). Penny and Jones calculated collision cross-sectional areas as the Space Shuttle cross-sectional area plus twice the average cross-sectional area of the debris objects in the volume. The authors' rationale for this calculation

revolved around the possibility of a debris object just glancing the Space Shuttle. As noted earlier, Penny and Jones also employed a different and somewhat simpler method in calculating the relative velocity between the Space Shuttle and the colliding debris object. However, several approaches to the calculation of collision probabilities did not include the relative velocity parameter at all, and these will be discussed in the next section.

Derivations without Relative Velocity Parameter. Grimminger's derivation of a collision probability equation that excluded a relative velocity parameter also excluded all other parameters except for the area of the SOI, the size of the debris, and the duration of exposure to the debris environment (8:21). Grimminger was interested in calculating the probability of a body being hit by a meteorite, and determined the average number of hits per hour by meteorites of specified size (n) to be:

$$n_1 = NA_b/24A_r$$

where

A_b = exposed body area

A_r = area of spherical surface (earth plus orbit)

The author calculated the average number of hits in time T as:

$$\bar{n} = T/\bar{t} = (NA_b/24A_r)T = n_1 T \qquad (8:21)$$

Grimminger used the above equation and the Poisson distribution to calculate the probability that a hit will occur exactly r times in time interval T, which is of the form:

$$P_r = \bar{n}^r \exp(-\bar{n})/r!$$

The probability that no hits will occur in Time T is therefore:

$$P_0 = p(0) = \exp(-\bar{n})$$

The probability that a hit occurs at least once in time T is:

$$p_{1+} = 1-\exp(-\bar{n}) \qquad (8:22)$$

This equation is virtually identical to those equations derived from the Poisson distribution except for the obvious deletion of several parameters considered important by other researchers.

Donald Kessler, Preston Landry, Burton Cour-Palais, and Reuben Taylor derived a collision probability equation identical to the Grimminger equation, although their approach was slightly different. The authors calculated the average number of impacts on the SOI as:

$$N = FAt$$

where

F = average debris flux (impacts/M^2-yr)

A = SOI area

t = time exposed to flux

They used this value in their probability calculation

of one or more impacts of the form:

$$P(\text{one or more impacts}) = 1 - \exp(-N) \qquad (18:38)$$

Chobotov also derived this same equation, again including only the debris flux, SOI area, and time while excluding the relative velocity parameter (2:484).

The final derivation effort to be discussed which did not consider the relative velocity parameter did not really follow any of the derivation methods discussed previously. However, the derivation is interesting because of its approach and because it is the only study found in the literature which used a Space Station as its SOI. Herbert Hecht conducted this study, considering a Space Station in a 270 nautical mile (500 kilometer) circular orbit at an inclination of 55 degrees, with a 10,000 m^2 cross-sectional area and a 10 year on-orbit time (10:Sec II, 1). Considering only trackable objects as candidates for collision with the Space Station, Hecht calculated the debris density in the Space Station flight path by selecting debris with apogees greater than 240 nautical miles and perigees less than 300 nautical miles (10:Sec II, 2). The author approached the problem two ways. First, he calculated the collision probability based on uniformly distributed debris populations in spherical shells. Second, he calculated the probability of

collision based upon debris dwell times within a toroidal band in which the Space Station traveled (10:Sec III, 1). More detailed descriptions of these approaches follow.

Derivation of the probability of collision using uniformly distributed debris in spherical shells involved calculating when the Space Station and debris reference areas overlapped as they swept along the proposed orbit. Hecht calculated that the average number of objects encountered in one orbit was simply the Space Station swept volume multiplied by the debris object density. This calculation could be extended to encompass the Space Station lifespan by using the orbital period of the station to recalculate the swept volume. Hecht felt the weakness of this approach was that it did not consider different dwell times of debris entering into the Space Station orbit (10:Sec III, 5). Consideration of debris dwell time in the Space Station orbit enabled Hecht to account for different durations of time in which particular debris was present in the Station orbital path. Hecht used a toroidal shell to facilitate the calculation debris dwell times in the collision probability derivation. The torus formed had as its central axis the Space Station orbital path (10:Sec III, 6). Debris dwell times were normalized with regard to their obital

periods, with actual dwell times for each trackable object being calculated using the Multiple Satellite Analysis Program. Hecht calculated the weighted presence for all objects in the torus as simply the sum of the normalized dwell times (10:Sec III, 8). The author then used this calculation to compute the weighted object density, and in turn multiplied this value by the Space Station swept volume to obtain the number of objects encountered over a specified time period. Hecht found that this value approximated calculations made using the Poisson distribution (10:Sec III, 11). However, this method, as did the first, did not consider the relative velocity parameter.

Conclusion

This literature review, as a whole, enabled me to better understand the dynamics of the space debris environment and the approaches taken to model that environment. The review also revealed what is known and unknown about the environment, as well as a bit of the synergism between system elements. In order to better understand this synergism and the ability of my model to represent that synergism, I constructed a causal diagram to conceptualize the space debris environment system. The next chapter covers the conceptualization portion of my model development.

III. System Conceptualization

Introduction

Despite the fact that my review of the literature provided me with an understanding of prior work on the space debris environment problem, conceptualization of the entire system continued to be important in the development of my model. The importance of this initial step in the system science paradigm lies in understanding the synergism between the elements that make up the system. System conceptualization complements my review of the literature by comparing my understanding of the dynamics of the space debris environment with that of other researchers. Furthermore, conceptualization of the system becomes the framework on which the actual parametric model is built and the results verified and validated.

I used causal diagram to aid in my conceptualization of the space debris environment system. This causal diagram, shown in Figure 3.1, presents pictorially all of the elements found in the system as suggested by previous research or by my own intuition. The arrows emanating from the system elements indicate the perceived positive (+) or negative (-) effect an increase of one particular element has on another element. Two-way relationships

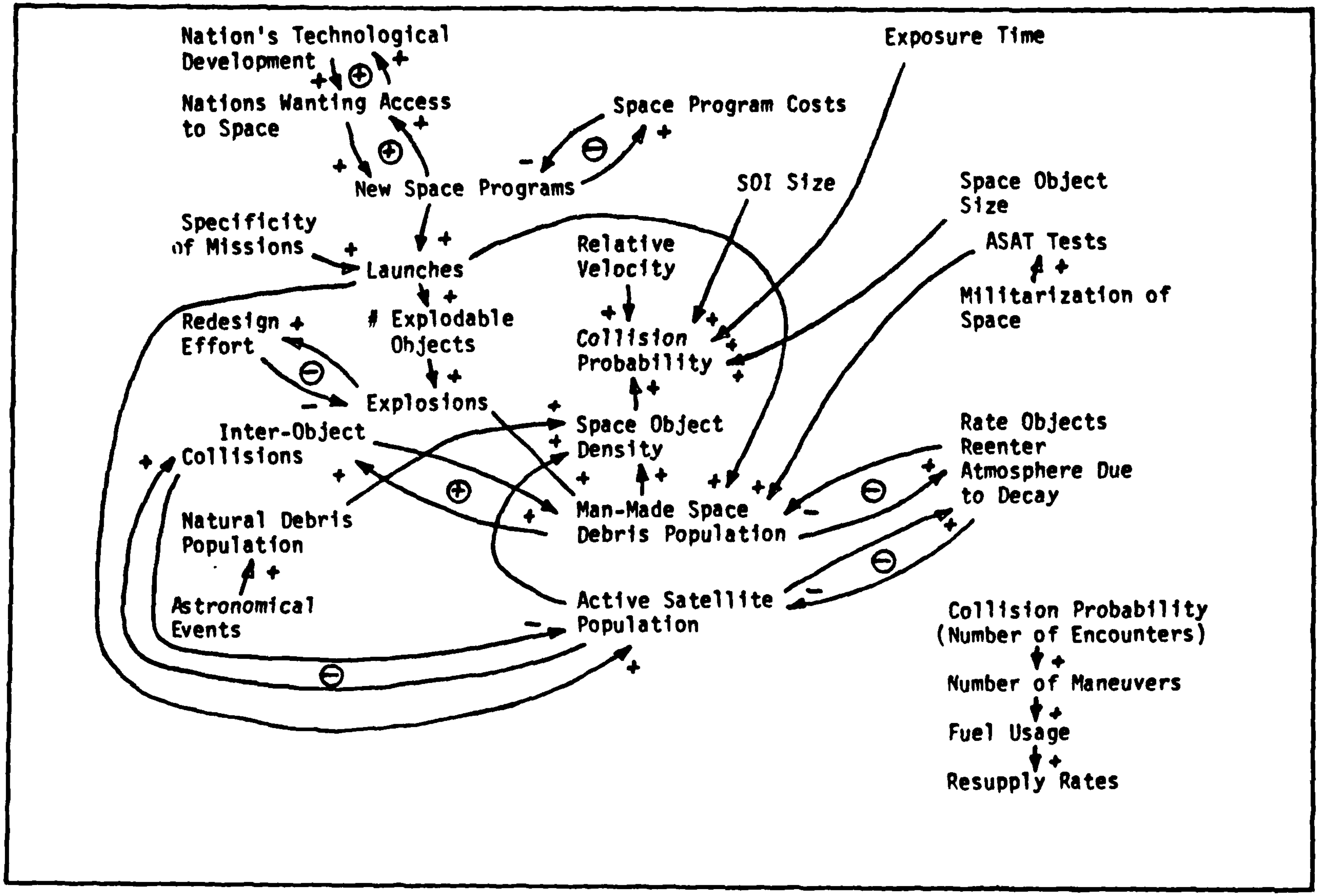


Figure 3.1. Causal Diagram

between elements form either positive (+) or negative (−) loops. A positive loop is self-perpetuating and causes instability in the system, while the negative loop is self-regulating and creates stability in the system. The remainder of this chapter discusses Figure 3.1 in more detail by concentrating on the relationships between each pair of system elements and the viability of those relationships as indicated by information found in the literature.

Causal Diagram Analysis

The objective of my research as stated in chapter one is to calculate collision probabilities and the number of encounters with debris requiring maneuvering by the planned Space Station and, in turn, use these calculations to determine the impact on Space Station fuel requirements. Therefore, the probability of collision and the number of debris encounters, which for causal diagram analysis purposes I will consider the same, are the components on whose interrelationships with the elements of the space debris environment system primary interest lies. The following paragraphs discuss each system element with regard to its relationship with other elements and with the probability of collision.

Exposure time to the space debris environment, the relative velocity between the SOI and the debris object

with which it will collide, the cross-sectional area of the SOI, and the space debris density were the parameters most commonly found in the collision probability calculations of other researchers as discussed in chapter two. The causal diagram shows that an increase in any of these parameters increases the probability of collision. The longer an SOI is exposed to the space debris environment, the more likely it is that it will collide with debris. Likewise, a higher relative velocity indicates that the debris will cross the path of the SOI more often, and hence have more opportunities to collide with it. An increase in the cross-sectional area of the SOI will cause the probability of collision to increase simply because the SOI will sweep out a larger volume of space where debris may be located. Finally, an increase in the spatial density of orbiting objects with which the SOI may collide will increase the probability of collision because there will simply be more objects available to collide with.

A parameter not considered in many derivations of the collision probability equation deals with the cross-sectional area of the debris. The causal diagram indicates that an increase in this parameter would have the same effect on the collision probability as the SOI cross-sectional area. Simply put, the SOI is more

likely to collide with large objects than small objects, given that the spatial density is the same. A determination as to the importance of this parameter will be made in chapter four during my derivation of the probability of collision equation.

The fact that several previous studies considered debris spatial density as the primary determinant of the probability of collision underscores the importance of including in the causal diagram the system elements that affect that spatial density. These elements consist of space launches, unintentional explosions, ASAT test explosions, inter-object collisions, the natural debris population, and the orbital decay of active satellites and debris into the atmosphere. The following paragraphs discuss each of these elements in more detail.

The causal diagram shows that several system elements exist that, while not directly affecting the debris spatial density, do contribute directly to the number of launches into space which is itself an important contributor to both the active satellite and debris populations. The 1983 TRW Space Log listed fourteen nations involved in sponsoring launches (32:120). As nations develop their technology, it is logical that their desire for access into space will increase, as has been the case with the United States

and the USSR. This increased desire will create more incentives to develop technology, forming a positive loop as indicated by the causal diagram. Again, the American and Russian space programs verify this condition. The desire and ability to use space will create new space programs, and alternatively, these programs will most likely generate more ideas and hence more desire to use space. The technological development, desire to access space, and the space program positive loops are tempered by the costs associated with the space programs, which the causal diagram indicates with a negative loop. While the number of new space programs may be constrained somewhat by cost, an increase in their *number* will likewise increase the number of space launches. Another element of the system affecting the number of launches in a positive manner is the specificity of the missions. In other words, the less a particular mission will accomplish in satisfying the objectives of new space programs, the more space launches will be required to satisfy those objectives. An increase in the number of launches, in turn, directly increases both the active satellite and debris populations, as discussed in chapter two. Finally, an increase in these populations naturally increases the spatial density of objects in orbit.

Unintentional explosions, as from defective spent boosters, and intentional explosions as a result of ASAT tests are the primary contributors to the space debris population. As the number of unintentional explosions increases and investigations into the reasons behind those explosions indicate ways to reduce or eliminate those occurrences, redesign efforts increase which create a stabilizing negative loop between the two elements. However, as the number of explosions from defective items still in orbit over which we have no control increases, the debris population and, likewise, the debris density will continue to increase. An increase in the number of ASAT tests also increases the debris density. The number of ASAT tests, in turn, is dependent on the intent of nations to militarize space. As Nicholas Johnson found in his research on the Soviet ASAT test program, this intent translates into increased debris densities.

Two additional sources of debris, inter-object collisions and the natural debris flux, are not major contributors to the debris spatial density but do contribute to the apparently destabilizing nature of the space debris environment system. The causal diagram shows that inter-object collisions and the space debris population form a destabilizing positive

loop. Of course, as inter-object collisions increase, we can expect the debris population to increase. As this population increases, however, the probability of elements of this population colliding increases also. The number of inter-object collisions forms a "stabilizing" negative loop with the active satellite population. As the number of inter-object collisions increases, the probability that a particular collision involved an active satellite other than the SOI increases, thereby decreasing the number of active satellites. As the active satellite population increases from launches, on the other hand, the spatial density of orbiting objects increases and thus the probability that two objects will collide increases. This negative loop, while stabilizing from a causal diagram perspective, is obviously destabilizing to those interested in the survivability of all active satellites.

The second minor contributor to the debris spatial density, the natural debris flux, primarily consists of meteorites traversing the orbits of active satellites and man-made debris. This flux is dependent on astronomical events such as the passing of the earth through the tail of a comet. An increase in the occurrences of these events causes the natural debris flux to increase above the normal levels, which in turn

causes the total debris spatial density to temporarily increase. As many researchers agree, the growth of the man-made debris flux above that of the natural debris flux combined with the potential controllability of the man-made flux lessens the importance of the natural debris flux in the collision probability problem.

The only system element at this time that contributes directly to the stabilization of the space debris population is orbital decay. The causal diagram shows that the rate at which objects reenter the atmosphere due to decay forms negative loops with both the active satellite and debris populations. The decay rate depends on the altitude of the object, the state of the atmosphere, and the object's size and density (29:121-127). The larger the proportion of objects having parameter values consistent with faster decay rates, the larger the decrease in both the active satellite and debris populations will be. While orbital decay does contribute somewhat to the stabilization of the orbiting object populations, its contribution is overwhelmed by the destabilizing contributions made by the sources of debris mentioned previously.

At this point I would like to discuss the relationship between the probability of collision and the Space Station fuel requirements using the causal

diagram. The diagram indicates a one-way, positive relationship between the probability of collision, the number of maneuvers required, the amount of fuel used, and the required resupply rates. An increase in the number of close encounters with debris will require that the Space Station perform more avoidance maneuvers, which in turn uses more fuel. Should this fuel usage exceed original plans, putting stress on the reserve fuel capacity, additional resupply missions by the Space Shuttle would be required. This sequence of events will directly increase the cost of the Space Station and Space Shuttle programs for the United States, and may indirectly increase the costs of other nations using these assets. Increased cost could constrain the realization of new space programs. Therefore, the causal diagram shows that a link exists between the probability of collision and the ability of man to use the resource of space.

Conclusion

Conceptualization of the space debris environment systems using a causal diagram has provided the framework with which to view the system as a whole and on which to actually build the parametric model. Due to the preponderence of positive loops and relationships between space debris environment system elements, it is apparent that the system is inherently

unstable. Inputs to the debris population overwhelm outputs, causing continuous growth within the system. Therefore, the development and subsequent analysis of the parametric model had to account for this instability. The next chapter presents a description of that model--its development, assumptions, and structure.

IV. Model Description

Introduction

Upon completion of a review of the literature and the conceptualization of the space debris environment system, I was now prepared to develop a parametric model for the purposes of obtaining preliminary collision probability calculations for the proposed Space Station and the number of trackable debris encountered requiring avoidance maneuvers. In this chapter I will first briefly describe the proposed Space Station and its associated parameters. I will then present the parametric model including the rationale behind its overall structure, a breakdown of each of its components, and the assumptions made in its parameters.

Space Station Description

Many people still picture the first space station as a single, monstrous facility housing hundreds of individuals. In reality, however, NASA plans to develop a Space Station "system" consisting of a number of separate manned and unmanned orbiting satellites. Current planning calls for a manned "core" element which will be the first element of the system deployed. This element will serve as the habitat for the astronauts assigned to the Space Station, and will

also contain research and development laboratory facilities, pilot production capability, servicing facilities for satellites and other space vehicles, logistics support for other elements of the Space Station system, and transportation capability to those elements.

Becoming operational in 1992, the Space Station core element will grow in both size and capability until it reaches maturity in the year 2000. The Space Station will be assembled and serviced by the Space Shuttle, with servicing missions occurring on a 90-day basis (26:132). The core element configuration currently favored by NASA engineers is the Power Tower or "T" configuration shown in Figure 4.1 (36:17). It appears that the core element will be deployed in a circular orbit at an inclination of 28.5 degrees and at an altitude of 500 kilometers, with the possibility of another core element being deployed at a later date at 400 kilometers altitude and 90 degrees inclination (34:23).

The remainder of the Space station system will consist of unmanned space platforms where scientific experiments and production facilities will be located. Astronauts from the core element will travel to these platforms for service and repair using orbital transfer vehicles (OTV's) (26:132). The evolution of the Space

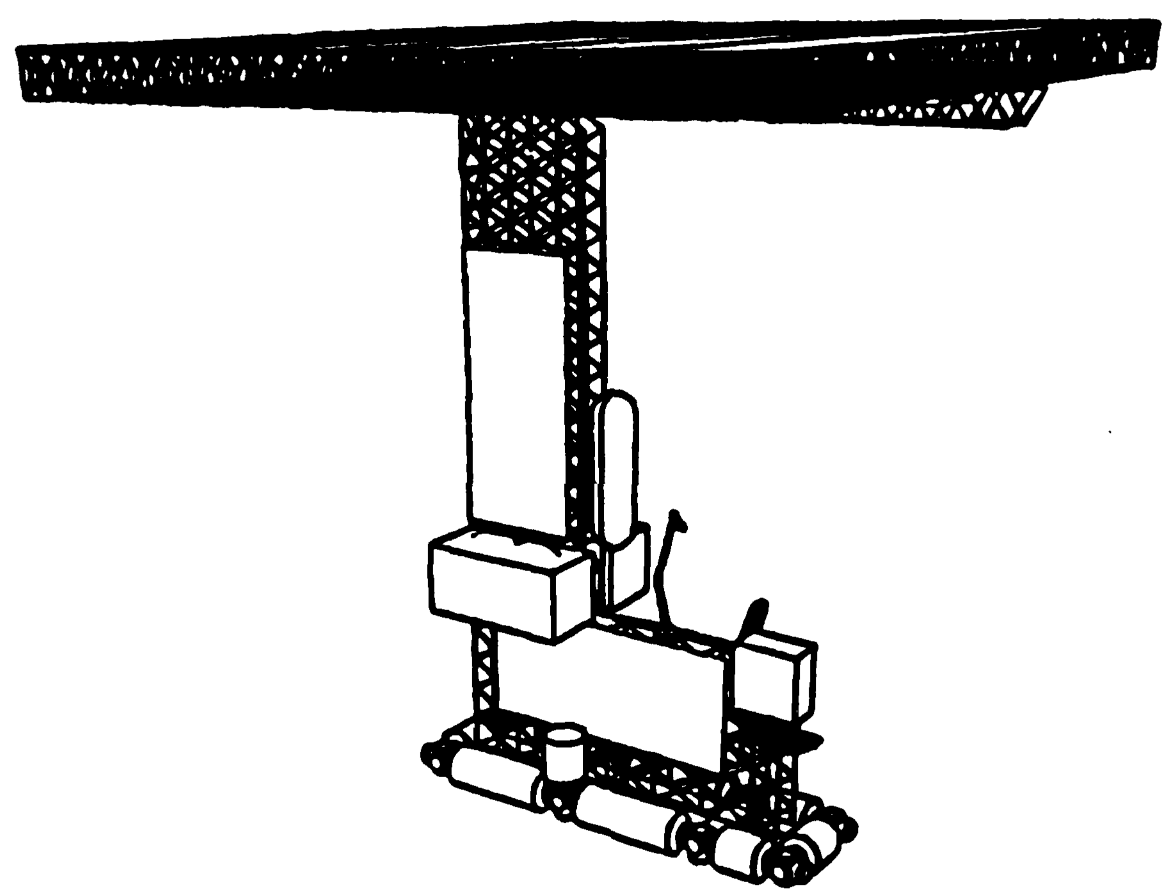


Figure 4.1. Space Station "T" Configuration

Station system may look something like that shown in Figure 4.2.

Overall Model Structure

Upon combining the information obtained concerning the space debris environment with that concerning the Space Station system, I realize the potential complexity in describing the dynamics of the debris environment in detail while involving all elements of the Space Station system. I decided that for the purposes of my analysis, a more top level, systems approach simulation model would be in order for several reasons. First, the many remaining unknowns in the space debris environment will be present regardless of the model's complexity. Second, a model that would keep track of all the parameters involved in calculating collision probabilities such as the orbital parameters of each orbiting object and its position at each point in time would indeed provide more accurate calculations for the current environment, but would be of no greater use in predicting the environment over the next thirty years. Therefore, a simulation model would provide the necessary flexibility to monitor certain system elements and perform the desired sensitivity analysis. Third, the object of my analysis deployed, and (3) the importance of its survival is

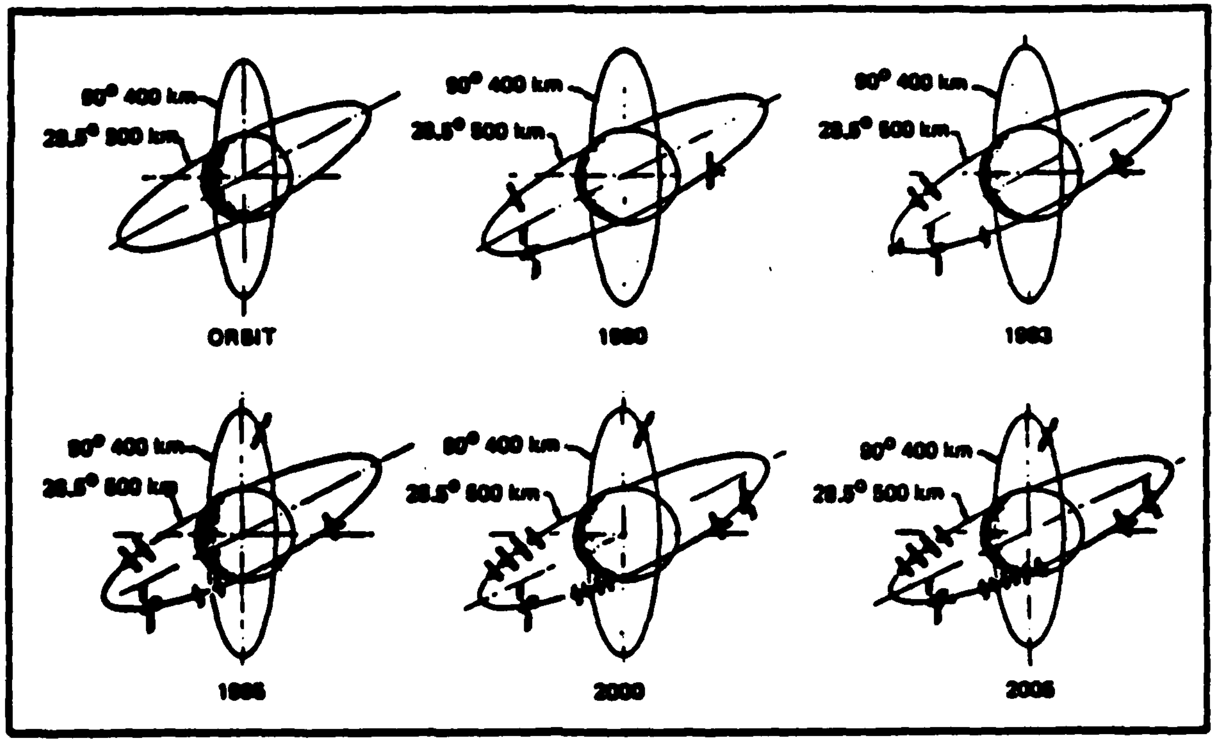


Figure 4.2. Space Station System Evolution

paramount because it will be manned, I was most interested in the interaction between the debris environment and this core element alone, which from this point on I will call the Space Station.

Based upon findings resulting from the literature review and upon the realization that model validation would require recent data, I decided to structure my model based on altitude only. My rationale was that I felt my results would be representative, since other researchers had found little difference in their results regardless of the dimension of model they used. Also, the use of certain parameters similar to the most recently developed model, that by Penny and Jones, would enable me to use the most recent data and assumptions shared by experts in this field.

I divided the debris population into three altitude bands, or concentric shells, surrounding the earth: 200 to 400 kilometers, 400 to 600 kilometers, and 600 to 900 kilometers. The system elements identified in the literature and during system conceptualization which I included in my model were launches, ASAT tests, orbital decay, unintentional explosions, and inter-object collisions. I felt the occurrence of these elements would significantly alter the debris populations which in turn would affect the calculation

of the Space Station collision probability and the number of encounters with tracked debris requiring avoidance maneuvers.

Due to my interest in future collision probabilities, I did not keep track of each object's position in space because (1) there was no way of knowing where the untrackable debris was located, and (2) there was no way of accurately predicting the exact location of future objects deposited in space. Consequently, I assumed the objects in each altitude band would be uniformly distributed. In addition, I assumed that the average cross-sectional areas of the objects within each altitude band remained constant, as well as the average orbital velocities. The rationale behind this assumption was similar to that for the first assumption, since there was no way of determining the actual parameters of future debris with any confidence.

The run length of the model was designed for 30 years, which starting in 1984, would provide data leading up to the deployment of the Space Station, during its growth to maturity, and for a period of time after it reached maturity.

The framework on which the parametric model was built involved discrete event simulation using the SLAM

simulation language. I took this approach because (1) all of the elements which I felt should be included in the model could be thought of as occurring at discrete points in time, and (2) simulation using a network did not apply since there is not a single entity traveling through the system. Discrete event simulation using SLAM involves determining the events where changes in the system can occur and then modeling each event type using Fortran subroutines. SLAM controls the scheduling of these events by putting them on an event calendar. When the simulation reaches a time that corresponds to a particular event, that event occurs (27:223-224).

The space debris environment system elements included in the simulation model were developed as individual subroutines. In addition, I included subroutines accomplishing the initialization of variables, the calculation of Space Station collision probabilities and the number of encounters with debris requiring avoidance maneuvers, the periodic check of system parameters, and the presentation of results. SLAM calls the initialization subroutine first, which initializes all variables and schedules each event on the event calendar for the first time. The simulation then proceeds for the desired run length, with events occurring in the order they appear on the event

calendar. After the ending time is reached, SLAM calls the output subroutine which presents the desired output. The entire parametric model can be found in Appendix A, with a list of variables and their definitions found in Appendix B. The following sections will now present each of the subroutines in detail.

Initialization Subroutine

The purpose of the INTLC initialization subroutine is quite obvious--to set initial values for all variables found in the simulation model. A flow diagram depicting the major portions of the subroutine is presented in figure 4.3. The simulation is set to start in 1984 and run for thirty years to the year 2013. I selected 1984 as the starting year despite the fact that the Space Station will not become operational until 1992 because I wanted the model to at least start with known parameters on which the system dynamics would depend. I next initialized all of the variables used in the calculation of average values over the number of simulation runs, which entailed initializing them only for the first of those runs. I then initialized the remaining array variables.

Setting the initial values for some of the more important system parameters came from various sources.

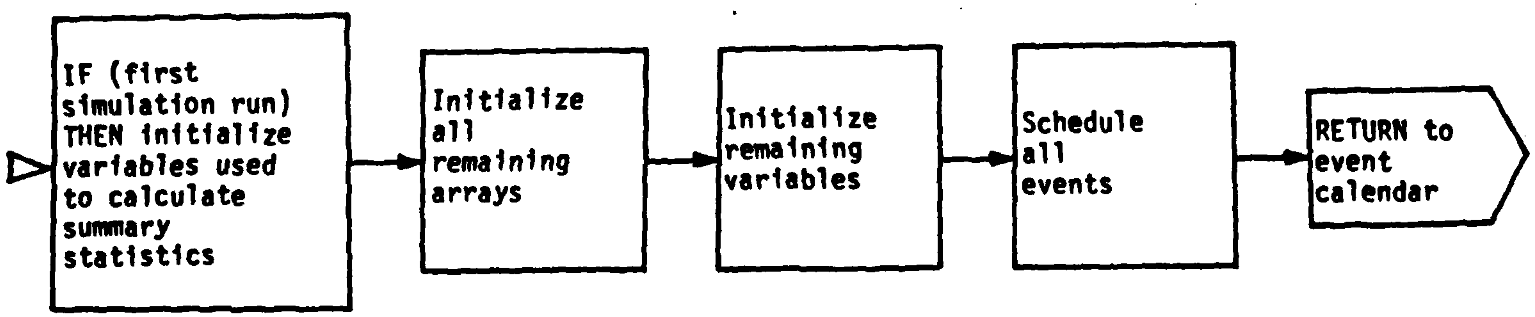


Figure 4.3. Initialization (INTLC) Subroutine Flow Diagram

I used schematics of both the initial and final Power Tower configurations for the proposed Space Station to calculate the cross-sectional area at IOC and at system maturity (19:48-49). I used only the side view of the Space Station because, according to Dr. Wiesel, most objects in LEO can be assumed to be in circular orbits. Therefore, one would expect the majority of debris to collide with the Space Station from the side, since NASA plans that the Power Tower configuration will maintain the orientation to earth depicted in Figure 4.4.

Since this model does not keep track of each orbital object's orbital parameters, I calculated average velocities for each altitude band to be used in the collision probability calculations. I used the circular orbital velocity equation:

$$v_c = (\mu/a)^{1/2}$$

where

a = altitude from earth's center (km)

μ = universal gravitational constant (km^3/sec^2)

I used this equation because of the assumption made earlier concerning the predominance of circular orbits in LEO. The average velocities calculated were simply the average of the same calculations made at the middle and boundaries of each altitude band. For example,

$$VELLO_1 = (398,601.3/6578)^{1/2} = 7.7843512 \text{ km/sec}$$

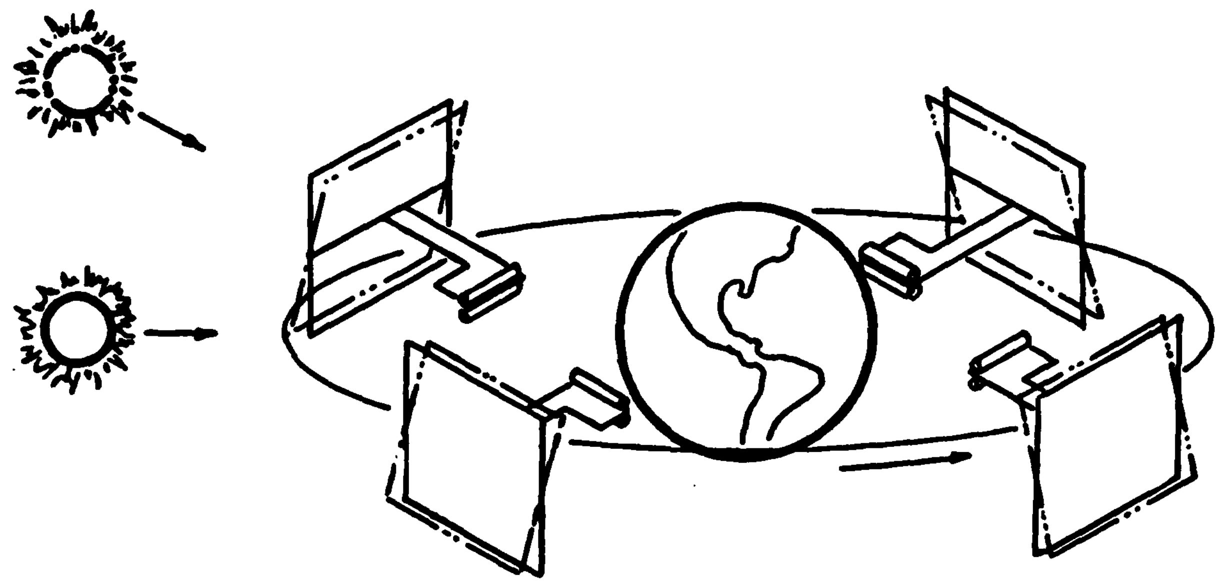


Figure 4.4. Space Station Orbital Orientation

$VELLO_2 = (398{,}601.3/6678)^{1/2} = 7.7258478$ km/sec

$VELLO_3 = (398{,}601.3/6778)^{1/2} = 7.6686439$ km/sec

$VELLO = (VELLO_1 + VELLO_2 + VELLO_3)/3 = 7.726281$ km/sec

I utilized the August 1984 CLASSY catalog to obtain the initial tracked debris populations for my model. For the altitude bands of interest, 2,593 objects were found, with approximately 12% found in the low altitude band, 32% in the medium band, and 56% in the high band (6). Penny and Jones had also used the CLASSY catalog to calculate the relative percentages of objects with an average radar cross-section (RCS) below 1.0 m^2. Assuming that these values had not changed appreciably in a year's time, I used the same values to calculate the number of trackable small objects (24:33). The number of large objects, those having an average RCS greater than 1.0 m^2, was simply the difference between the total trackable object population and the trackable small object population for each altitude band.

Since I was interested in including the untrackable debris population in my collision probability calculations, I needed to readjust the small object as well as the total populations. I assumed the untrackable population as being three times as large as the trackable population, which was taken from the initial estimate made by Penny and Jones as a

result of survey results obtained from experts in the space debris environment field (24:33). Therefore, I recalculated the total and small object populations for each altitude band.

In entering the dynamics of orbital decay into the space debris environment model, I used decay constants consistent with those used by Penny and Jones. The values represent the percentage of objects decaying out of a particular altitude band in one week's time (24:34). Since the largest percentage of objects is found within the altitude bands I am considering, and since objects at higher altitudes take literally hundreds of years to decay, I assume that for the length of time my model operates, no objects decay from higher altitudes into the high altitude band.

Unable to obtain concrete information as to the actual number and distribution of explodable objects in space, I was left to rely on certain estimates by experts. I used the survey response obtained by Penny and Jones of 1400 total explodable objects in low earth orbit as the basis for determining the number of explodable objects to be used in my model. Since the total trackable population for my altitudes of interest is 50.9% of the total trackable population as listed in the CLASSY catalog, I assumed that that same proportion could be applied to the explodable object popluation.

Therefore, I calculated 713 explodable objects to be in orbit between 200 and 900 kilometers. As a result of an interview with Dr. Donald Kessler, I followed his assumption that the number of explodable objects at a particular altitude was proportional to the total number of objects in that altitude band (16). Therefore, I multiplied the total explodable object population calculated above by the relative percentages of objects in each altitude band to obtain the number of explodable objects in that particular band. For example:

EXPLO = EXPOBJ x RELLO = 713 x (1244/10372) = 85

The final function of the initialization subroutine is to place all of the events once on the event calendar so that the model can continue their scheduling through the remainder of the simulation time. To accomplish this, the time between occurrences of a particular event, or the inter-arrival time, needed to be determined. Since I was using orbital decay constants based upon a one-week period, I scheduled the subroutine responsible for readjusting the debris populations due to orbital decay on a weekly basis. I also decided to calculate the yearly probability of collision between debris and the Space Station on a weekly basis beginning in 1992. I felt that this sample of 52 over a year's time would be

representative of the changes in the calculation due to debris density changes throughout the year. I scheduled the yearly accumulation of data over the desired number of simulation runs to occur on a yearly basis beginning in 1984. Remembering Kessler's prediction that inter-object collisions would become the primary contributor to the debris population in the future, and not knowing the magnitude of these occurrences even at the present time, I decided initially to make a daily check for inter-object collisions within any of the altitude bands. This value could be readjusted later should verification of my model show more collisions occurring than allowed for.

The remaining events involve system elements which are sources of debris: launches, ASAT tests, and unintentional explosions. These events were scheduled with the realization that results of the model could possibly be greatly affected by unrealistic inter-arrival times for these events. According to Dr. Kessler, the current total number of launches lies in the range between 120 and 150, and analysis of data found in the 1983 TRW Space Log verified these values (16;32). Therefore, I used these values as minimum and maximum launch rates and used 135 launches as the yearly mean rate. Since it is impossible to accurately

estimate a growth rate in future launches, the parameters listed above were used throughout the simulation run. I used the exponential distribution with these parameters to generate inter-arrival times between launches because its "memoryless" property describes the independent scheduling of launches among all nations. Research by Nicholas Johnson on the Soviet ASAT test program and Penny and Jones survey results lead to the estimate of one to three ASAT tests by all nations per year being included in the space debris environment model. An exponential distribution with a mean of two ASAT tests per year was selected to generate inter-arrival times between the occurrence of these tests. Again, like launches, ASAT tests are assumed to occur independently of one another. As for launches, this test rate was used throughout the simulation run because it was impossible to predict test rates in the future with any accuracy. The sources of debris that are the most unpredictable in their occurrences are unintentional explosions. Individuals knowledgeable in this area responded to the Penny and Jones survey by estimating that one out of every 500 explodable objects exploded each year. I used this parameter as the mean of an exponential distribution to generate the time between explosions since the "memoryless" property again described the

independent nature of these occurrences. After the initial scheduling of the unintentional explosion event as well as all of the other events, the SLAM control language moved to the event calendar to generate the occurrence of events and schedule future events for the balance of the simulation run.

Event-Scheduling Subroutine

The event-scheduling subroutine, EVENT, is required when using SLAM discrete event simulation to call the appropriate event subroutine whose calendar time matches the simulation time, designated TNOW. Figure 4.5 is a flowchart depicting the major elements of this subroutine. As the figure shows, all of the event subroutines included in the model are available to be called at the appropriate time except for the initialization and output subroutines, which are automatically called by SLAM at the beginning and end of the simulation run, respectively.

ASAT Test Subroutine

The ASAT test subroutine, simply entitled ASAT in the simulation model, involves the occurrence of debris-depositing ASAT tests and the distribution of that debris in the altitude band of test occurrence. The flow diagram pictured in Figure 4.6 indicates the major functions performed by this subroutine--the

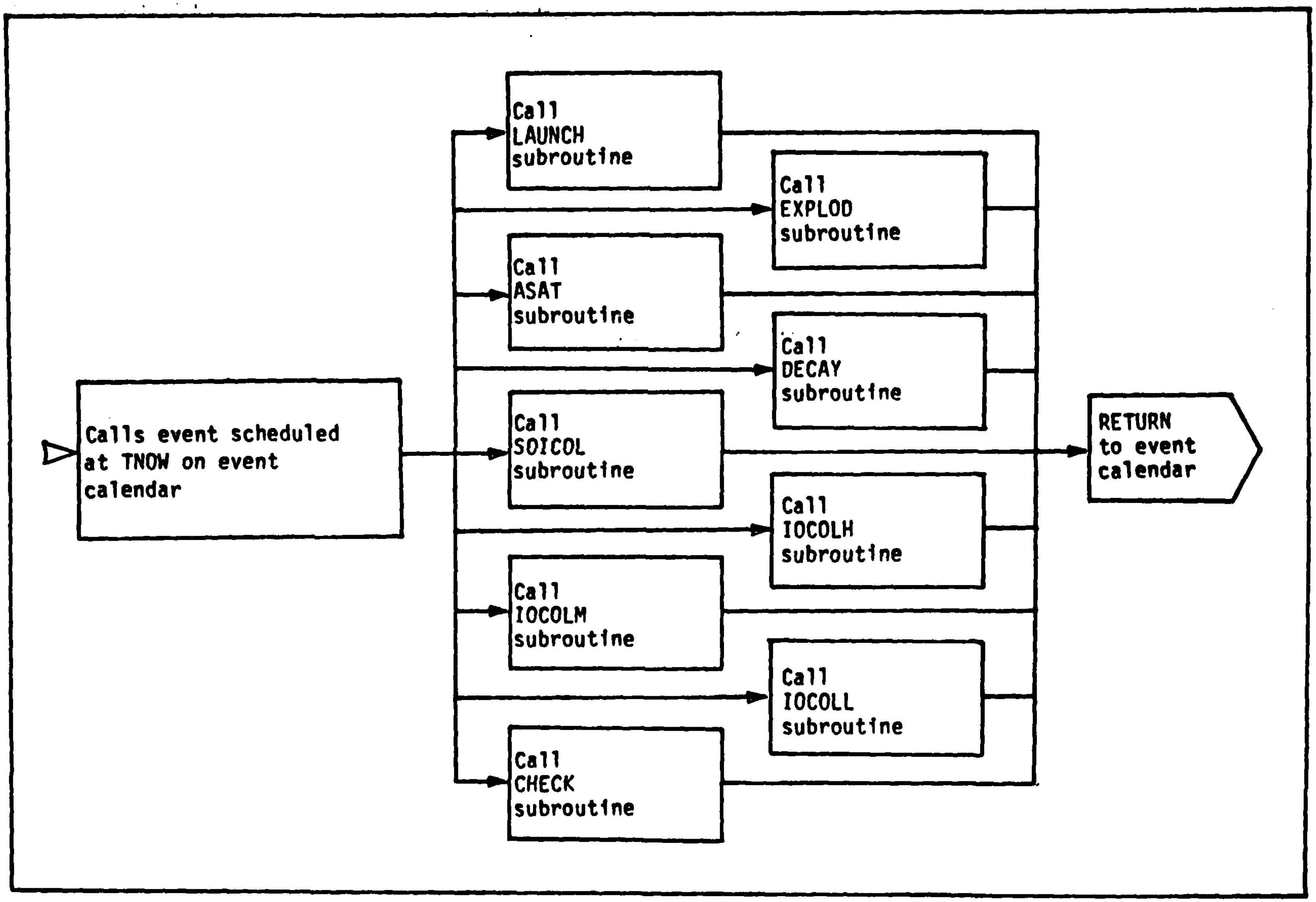


Figure 4.5. Event-Scheduling (EVENT) Subroutine Flow Diagram

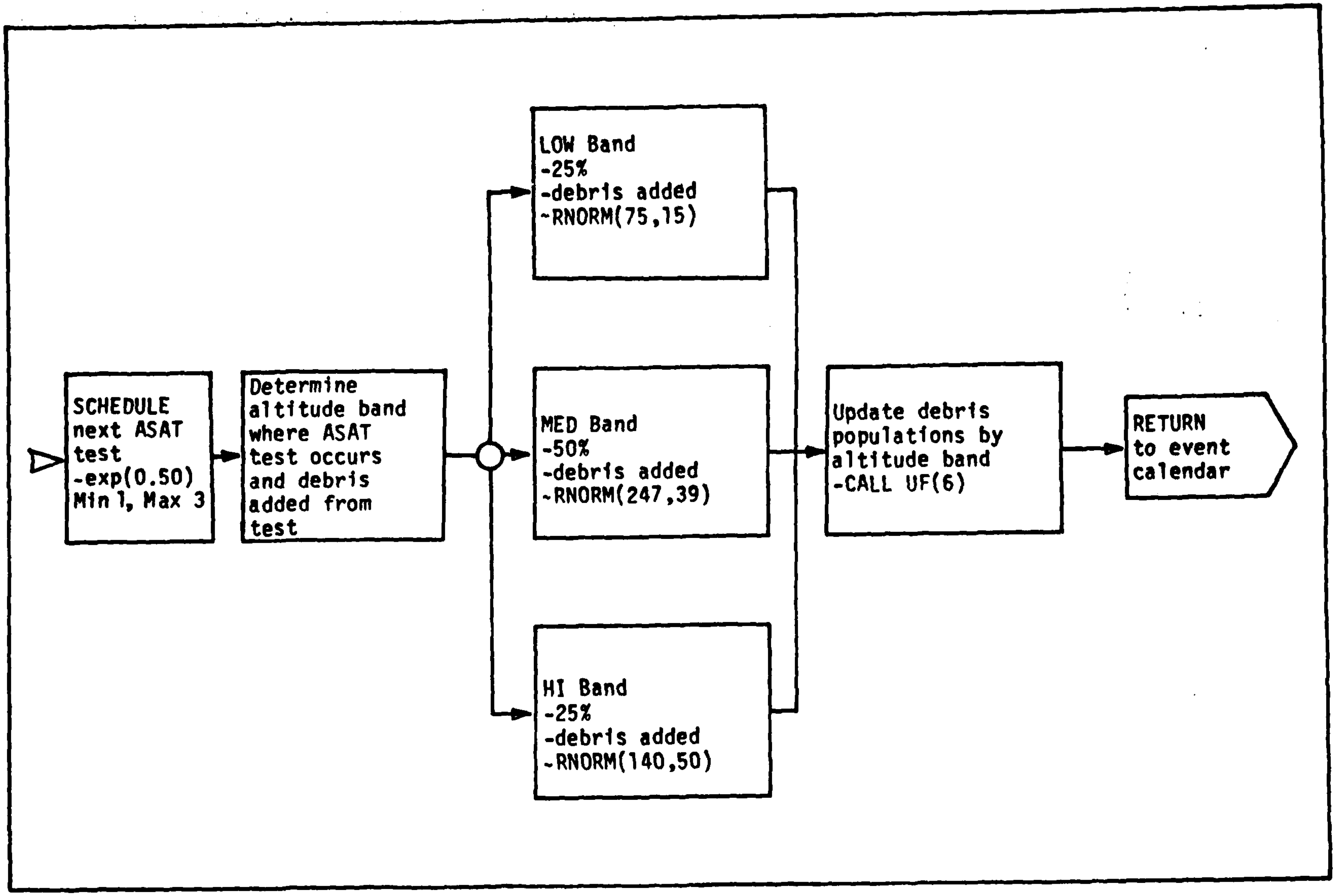


Figure 4.6. ASAT Test (ASAT) Subroutine Flow Diagram

scheduling of the next ASAT test, determination of the altitude band where the ASAT test occurred, determination of the quantity deposited in that altitude band, and the updating of the debris population as a result of this test. As presented earlier during the discussion of the initialization subroutine, the scheduling of the next debris-depositing ASAT test involves using an exponential distribution with a mean of two ASAT tests per year to generate an inter-arrival time which is added to TNOW and placed on the event calendar. The following paragraphs discuss the remainder of the ASAT subroutine functions.

I determine the altitude band where the ASAT test occurs by first using a uniform distribution between the values of 0.0 and 1.0 to generate random values with which to determine the appropriate altitude band. Research by Nicholas Johnson indicated that, historically, 50% of the debris-depositing Soviet ASAT tests occurred in what I am defining as the medium altitude band, with 25% occurring in each of the other two altitude bands (12). I differentiate between those ASAT tests which deposit debris and those that do not simply because only debris-depositing tests have an impact on the debris densities. Not being able to obtain data on the United States ASAT test program left

me to assume that the Soviet ASAT test distribution is representative of all ASAT tests.

I next determine the quantity of debris deposited in that band. According to an article on the fragmentation of asteroids and artificial satellites written by Dr. Wiesel, resulting debris does not separate from the collision altitude by more than 20 kilometers (38:114). Therefore, I assume that the debris generated stays in the same altitude band where the ASAT test occurred. The quantity of debris generated follows a normal distribution, as put forth by respondents to the Penny and Jones survey (24:107). The parameters for this distribution were obtained from an analysis of the historical data collected by Johnson on the Soviet ASAT test program. Again, for lack of data on other ASAT tests, these values were assumed to be representative of all ASAT tests. Before updating the debris populations, I keep track of the number of ASAT tests occurring in each altitude band per year by incrementing one of the variables ALRT, AMRT, or AHRT, depending on which altitude band the test just occurred in.

The last function of the ASAT test subroutine is to update space debris system parameters based upon the debris generated. This involves calling the sixth user function (UF(6)), whose basic function is to update

system parameters as a result of event occurrences as well as calculate the Space Station probability of collision. Within the ASAT test user function, the number of debris generated from a test is added to the debris population of the altitude band where the test occurred. The total debris population is then recalculated by simply adding the populations from each band together. I then determine the number of small and large objects generated from the ASAT test. I assume, as did Penny and Jones, that 98% of the debris generated is small due to the presumed high intensity of the explosion and the tendency to fragment both the interceptor and the target (24:40). The small and large object populations in the altitude band of occurrence are updated according to this assumption.

Orbital Decay Subroutine

The DECAY subroutine actually performs two major functions, as indicated by its flow diagram in Figure 4.7. First, it updates the debris populations in each altitude band weekly to account for orbital decay. Second, because this subroutine is scheduled on a weekly basis, I chose to include several statements that sample from the det is and explodable object populations in order to obtain averages of these populations over the year.

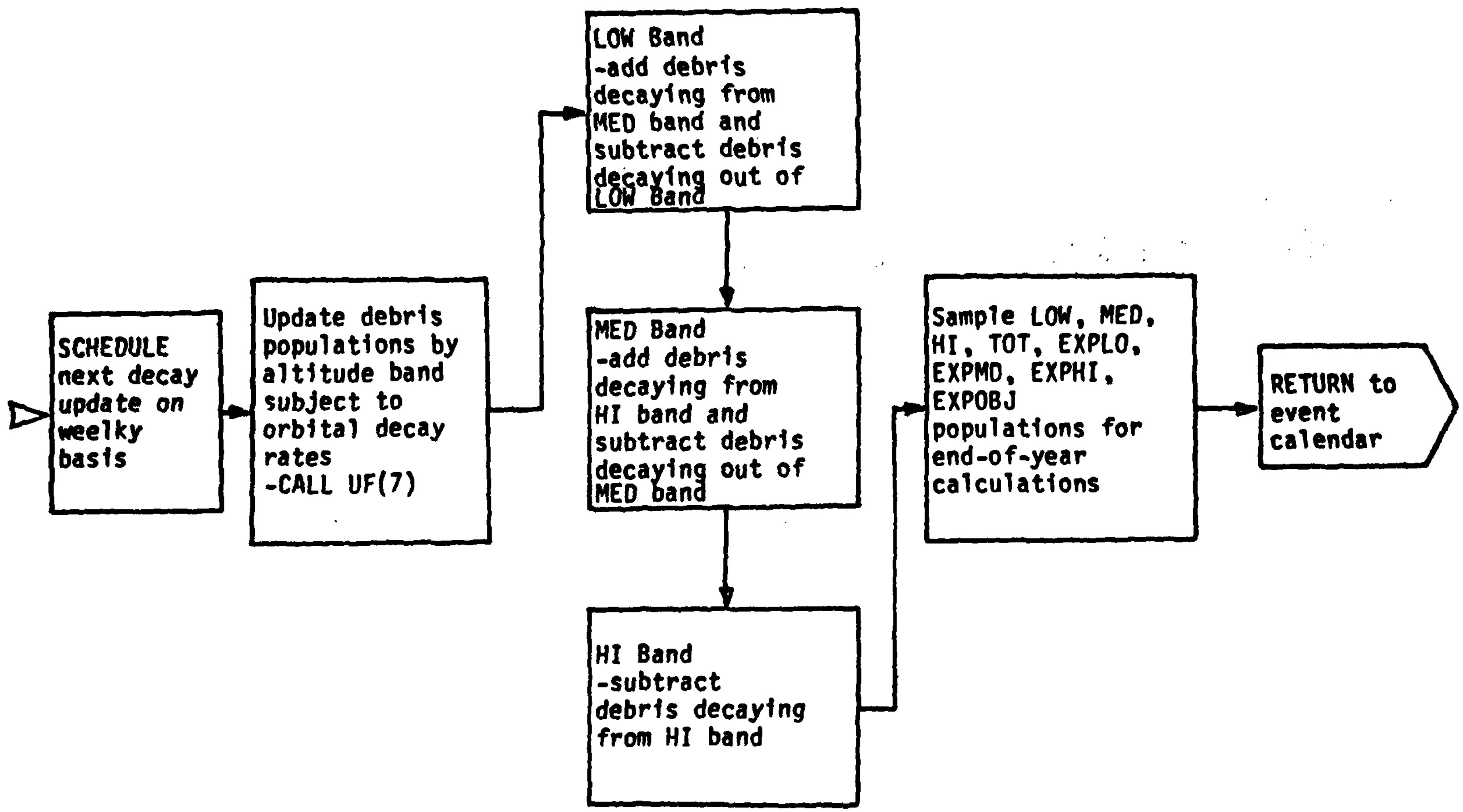


Figure 4.7. Orbital Decay (DECAY) Subroutine Flow Diagram

The orbital decay subroutine first schedules the next occurrence of this event on a weekly basis. The subroutine then calls UF(7), which performs the actual updating of the debris and explodable object populations by altitude band subject to the orbital decay rates established in the initialization subroutine. The low altitude band debris population is updated by adding debris decaying from the medium altitude band and subtracting the debris decaying out of the low band. I update the medium altitude band debris population by adding the debris decaying from the high altitude band and subtracting that debris decaying out of the medium altitude band. Finally, I recalculate the debris population in the high altitude band by subtracting out the debris decaying from that band. As mentioned during the discussion of the initialization subroutine, I assume that debris higher than 900 kilometers decays at too slow a rate and in too few numbers to impact the population in the high band for the run length of this simulation model. I use the same methodology as above to update the explodable object populations in each altitude band.

The second function of the DECAY subroutine consists of simply adding the current debris and explodable object populations to their respective arrays containing the summation of these populations

over the years of interest. In essence, I take note of and record these populations on a weekly basis. The summation arrays are later used in the CHECK subroutine to calculate the averages of these populations for each year the model collects data.

Unintentional Explosion Subroutine

The subroutine EXPLOD, whose flow diagram is presented in Figure 4.8, handles the time, location, and dynamics of an unintentional explosion in space. The primary functions of this subroutine are to (1) schedule the next unintentional explosion, (2) determine the altitude band where the explosion occurred, (3) determine the quantity of debris generated from the explosion, (4) decrement the appropriate explodable object population to account for the explosion, and (5) update the debris populations based upon the debris added. I discussed the method of scheduling the next unintentional explosion while reviewing the initialization subroutine, where the estimation that one out of every 500 explodable objects exploded each year was combined with the exponential distribution to produce explosion inter-arrival times. Therefore, the more explodable objects there are, the shorter the time between explosion events. A statement added into the subroutine insures that, should the opposite case be true where there are no explodable objects left, a future explosion event is not scheduled

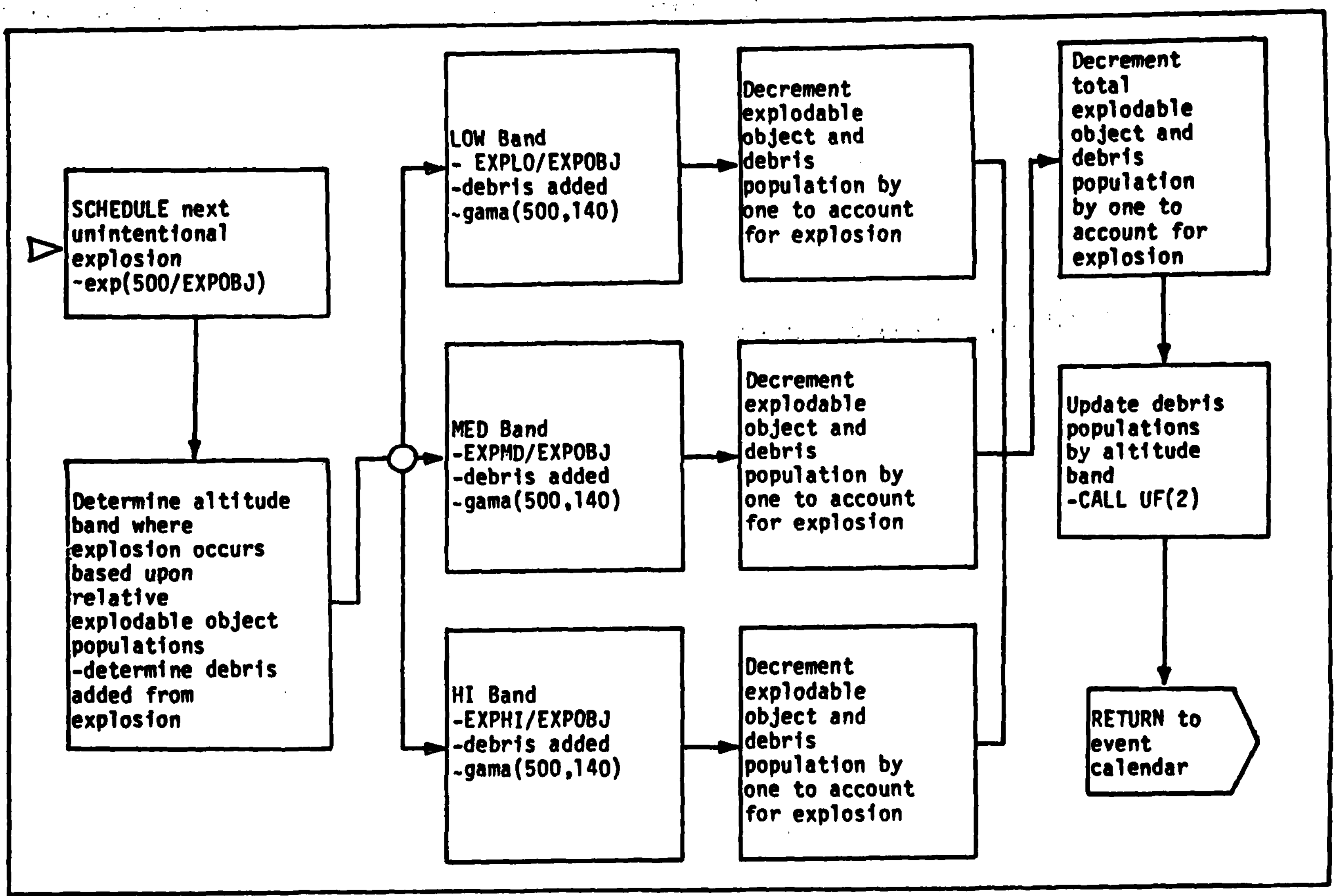


Figure 4.8. Unintentional Explosion (EXPLOD) Subroutine Flow Diagram

until a launch deposits explodable objects into one of the altitude bands. I will discuss the remaining functions performed by the explosion subroutine in the following paragraphs.

I base the decision as to which altitude band the explosion occurs on the relative percentage of the number of explodable objects in a particular altitude band to the total number of explodable objects. For example, the chance that an explosion would occur in the low altitude band should it be scheduled to occur at the very beginning of the simulation run is:

EXPLO/EXPOBJ = 85/713 = .1192146 or 11.92%

The rationale behind this determination is simply that an explosion has a greater chance of occurring where there are more objects. As in the ASAT test subroutine, I use a uniform distribution to produce random variates between the values of 0.0 and 1.0 to determine where the explosion will occur. After that determination, I calculate the amount of debris resulting from the explosion.

The selection of the gamma distribution to generate the amount of debris is based on survey results obtained by Penny and Jones (24:102-103). Also coming from that survey are the parameter values of 500 objects for the mean and 140 objects for the standard deviation. The minimum and maximum values obtained

from the survey of 5 and 15,000 objects, respectively, reflect the wide range of intensities for these explosions and the inability to describe the dynamics of this event more accurately. I assume that all objects generated from an explosion have no potential to explode themselves. I base this on the belief that the object that just exploded would be sufficiently fragmented to disallow the possibility of debris having the structural integrity to be potentially explodable.

The decrementing of the number of explodable objects in that altitude band where the explosion occurred by one acknowledges the elimination of that object as being potentially explodable. The object population in the same altitude band is also decreased by one to account for the loss of the explodable object. In order to keep track of the number of explosions occurring each year, variables named ELRT, EMRT, and EHRT are incremented by one each time an explosion occurs in their respective altitude band. Finally, the total explodable object and debris populations are decreased by one.

The last function performed by the EXPLOD subroutine consists of calling UF(2) which updates the debris populations to account for the debris resulting from the explosion. The debris is added to the debris population in the altitude band where the explosion

occurred, again assuming that all debris stays in the band in which it was generated based on research results by Dr. Wiesel. I make the assumption that 95% of the debris generated is smaller than 1.0 m^2 RCS, and adjust the small and large object populations within the altitude band of interest accordingly.

Orbital Launch Subroutine

The orbital launch subroutine entitled LAUNCH performs the particular functions associated with a launch of a spacecraft. These functions, presented in the flow diagram pictured in Figure 4.9, include determining the altitude band where the spacecraft enters into orbit, the amount of debris deposited from the launch, the number of new potentially explodable objects added to the environment due to the launch, and the updating of the debris populations. The subroutine also schedules the next launch using an exponential distribution with a mean of 135 launches per year, as presented earlier in this chapter. The remaining functions listed above are discussed in more detail below.

Determining the altitude band where the payload enters into orbit is a function of a random variate generated using a uniform (0.0, 1.0) distribution and the percentage of payloads that have historically been

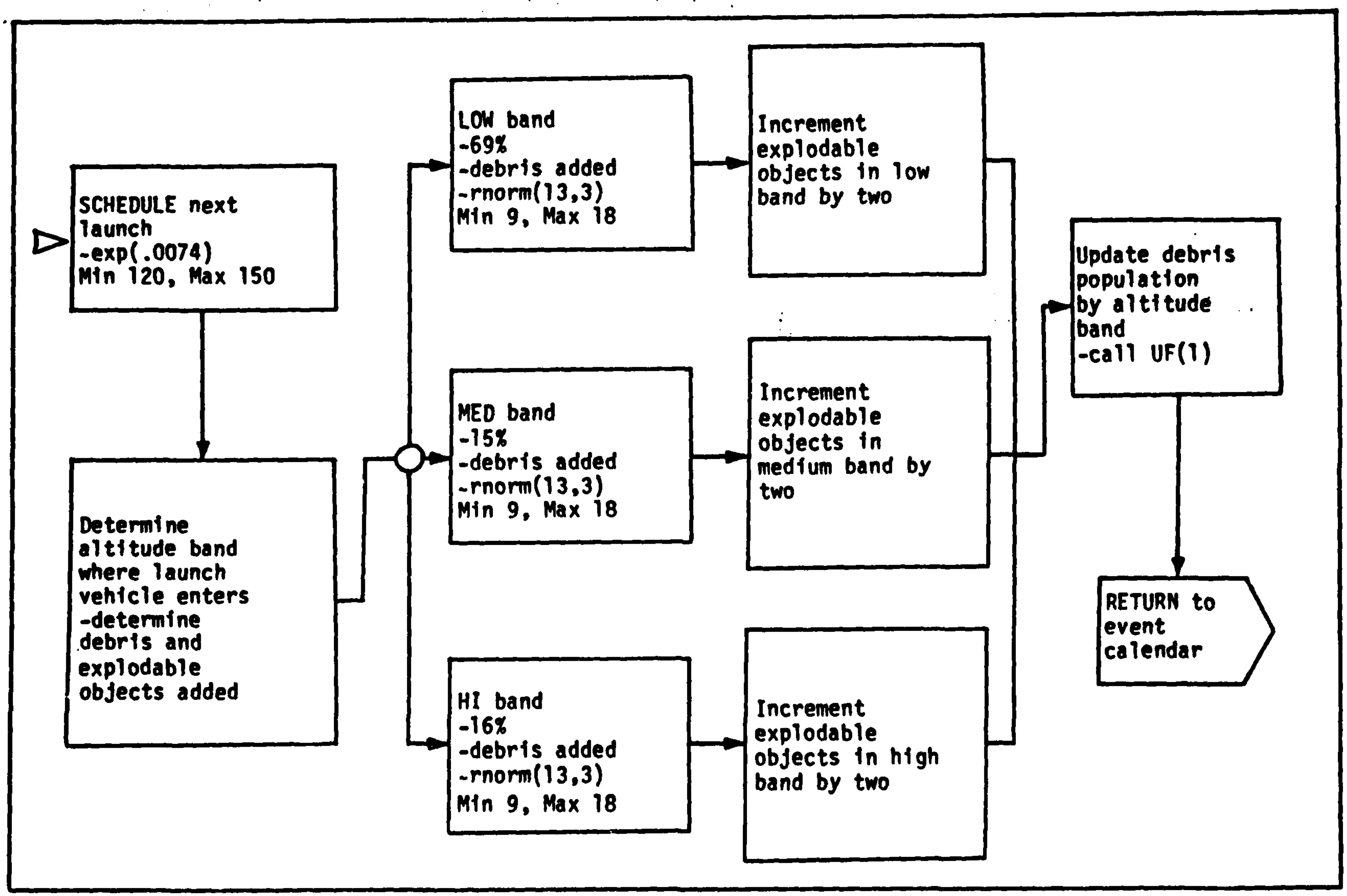


Figure 4.9. Orbital Launch (LAUNCH) Subroutine Flow Diagram

launched into the altitude bands selected for this model. Analysis of data from the TRW Space Log indicated that, for those launches targeted at the altitude bands of interest, approximately 69% of the payloads were put in the low altitude band, 15% were put in the medium altitude band, and 16% were put into the high altitude band (32). Upon determining where the payload was deposited, I next calculate the amount of debris and explodable objects generated from the launch. According to Dr. Wiesel, all debris generated lie in the same altitude band as the payload (37). I use a normal distibution with a mean of 13 and a standard deviation of three, which came from the Penny and Jones survey, to generate the amount of debris (24:97-98). Also, I assume no less than 9 and no more than 18 objects can result from a launch (24:97). According to the survey, each launch deposits two explodable objects in the same altitude band as the payload (24:99). Before updating the debris population, I increment the launch rate in the appropirate altitude band for the current year. The final function of the orbital launch subroutine is to update the debris population in the altitude band where the payload entered, which is done in UF(1). In assigning the newly added small objects to the small object populations, I assume that 90% of the total

number of objects generated by the launch are small.

SOI Collision Probability Calculation Subroutine

The subroutine responsible for calculating the Space Station collision probability and the number of encounters with debris is called the SOICOL subroutine, and is pictured in Figure 4.10. The subroutine itself schedules the next calculation on a weekly basis, and then calls UF(8), which actually performs these calculations. The procedures for these calculations follow.

I selected the Poisson distribution to calculate the Space Station collision probabilities since this distribution appeared to be appropriate for the situation present in my model. Hillier and Lieberman, in describing the Poisson distribution, stated that:

> "Heuristically speaking, this distribution is appropriate in many situations where an 'event' occurs over a period of time, like the arrival of a customer; when it is as likely that this 'event' will occur in one interval as in any other; also the occurrence of an event has no effect on whether or not another occurs." (11:339).

With respect to the space debris environment model, the "event" can be thought of as the collision between a debris object and the Space Station. Given small enough intervals of time where the debris densities do not change, the occurrence of a collision in these

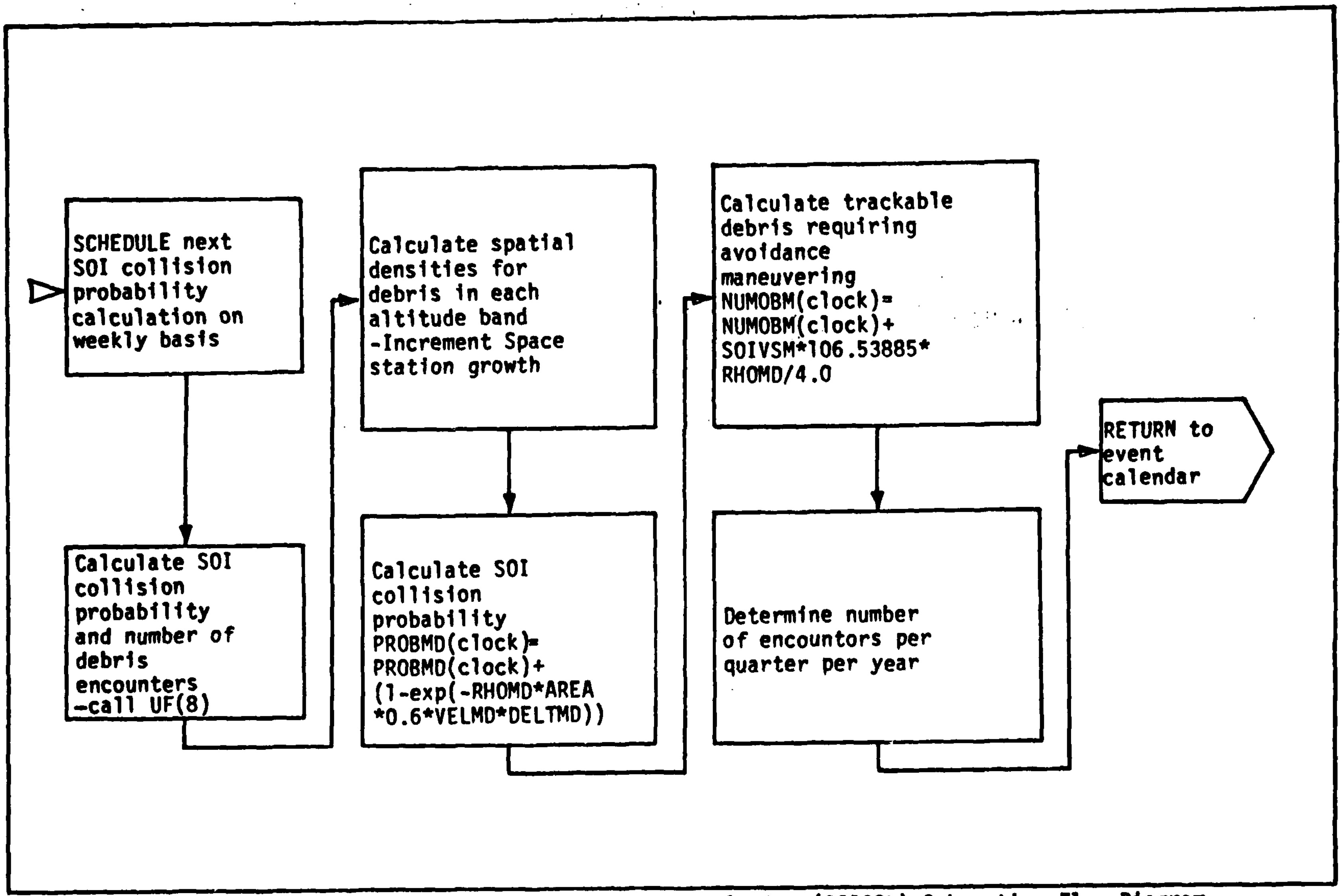
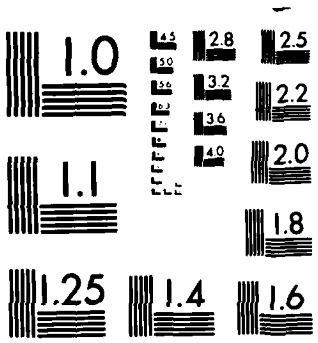


Figure 4.10. SOI Collision Probability Calculation (SOICOL) Subroutine Flow Diagram

AD-A151 872 ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 2/3

UNCLASSIFIED B M WAECHTER DEC 84 AFIT/GOR/OS/84D-15 F/G 22/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

intervals of time is equally likely. Finally, the collision with one object in no way affects the possibility of the Space Station colliding with another object. Therefore, the Poisson distribution is appropriate to use in the calculation of collision probabilities.

The general form of the Poisson distribution is:

$$P(X=k) = P_X(k) = [\lambda^k \exp(-\lambda)]/k! \quad (11:339)$$

where

λ = mean rate of "event" occurrence (positive constant)

k = number of "events" (nonnegative integer)

In order to be consistent, for comparison purposes, with other individuals who have used the Poisson distribution to calculate collision probabilities, I set out to calculate the probability of one or more collisions with the Space Station over a year's time. The derivation of the appropriate equation is as follows:

$$P(X=0) = [\lambda^0 \exp(-\lambda)]/0! = \exp(-\lambda)$$

$$P(X\geq 1) = 1 - P(X=0) = 1-\exp(-\lambda)$$

Now that we have the desired form of the equation, the only remaining unknown is the mean rate of "event" occurrence, λ. Since an "event" can only occur when an object and the Space Station are at the same place at the same time, this is equivalent to determining the

number of objects found within the volume swept out by the Space Station over a year's time. Obviously, this value is a function of several parameters: the debris spatial density, the Space Station cross-sectional area, the relative velocity between the colliding debris and the Space Station, and the length of time the Space Station is exposed to the environment.

The debris spatial density enters into the determination of λ simply because the greater the density, the higher the number of objects will be that lie within the Space Station's path. Of course, the larger the cross-sectional area of the SOI the greater the swept volume will be, and subsequently the greater the number of objects lying within this larger volume. Although the area of debris objects would also affect the parameter λ since the larger the area of each object, the more space the debris takes up and hence the greater likelihood it will lie within the swept volume, I assume that this is not a factor because the area of the debris is inconsequential when compared to the volume of the altitude band in which they preside. In other words, the average size of debris would have to be many times larger to be as critical to the collision probability calculation as the quantity of debris. The relative velocity, on the other hand, is an important factor in calculating λ since it involves the severity of

the collision. Since I am not accounting for each object in the debris environment, I make the assumption as proposed by Dr. Wiesel that most collisions occur at relative velocities equivalent to 60% of the SOI circular orbital velocity. At the altitude and inclination of the Space Station, there are very few objects that present any danger of a head-on collision. Therefore, most collisions occur at an angle more to the rear of the SOI. The final factor involved in the calculation of λ is the time of calculation. Obviously, the longer the period of time over which we determine the number of objects lying within the swept volume, the greater the amount of volume the Space Station sweeps out, and thus the more objects we are likely to encounter. Therefore, the parameter can be written as:

$$\lambda = \rho A(0.6 \times v)t$$

where

ρ = debris spatial density (objects/km^3)

A = Space Station cross-sectional area (km^2)

v = Space Station circular orbital velocity (km/sec)

t = time of measurement (sec)

Writing the same equation using the appropriate units for each of the factors, we can see that λ becomes the number of objects found within the swept volume over

the designated period of time:

$$\lambda = (\#\ \text{objects/km}^3) \times (\text{km}^2) \times (0.6 \times \text{km/sec}) \times (\text{sec})$$
$$= \#\ \text{objects}$$

The overall Space Station probability of collision calculation therefore becomes:

$$P(X \geq 1) = 1 - \exp(-\rho A 0.6 v t)$$

The SOICOL subroutine, through UF(8), updates each of the parameters used in the above calculation to correspond to system parameters found at that point in time. I recalculate the debris spatial densities using the current debris populations from each altitude band. Although I calculate the probability of collision on a weekly basis, I use the number of seconds corresponding to one year as my value for the time variable. This corresponds to calculating each week the yearly probability of collision given that the system parameters at that time were those found at the end of the year. This results in 52 samples of the collision probability being taken each year, so that an average can be calculated which acknowledges the changing system parameters over a year's time. I increment the cross-sectional area of the Space Station from 1992 through the year 2000 to correspond to the planned growth in the Space Station (19). The value of 0.0000004 is the weekly increase in growth, which I

assume to be constant over the eight-year period. The circular orbital velocity for the Space Station at an altitude of 500 kilometers was obtained using the following equation:

$$V_c = (\mu/a)^{1/2} = (398601.3/6878)^{1/2}$$
$$= 7.6126922 \text{ km/sec}$$

Since the time period of calculation and the velocity parameters remain constant, only the area of the Space Station and the debris spatial density vary. Once the spacecraft reaches system maturity, only the debris density ultimately determines the collision probability. This observation follows the conclusions made by other researchers, as discussed in chapter two.

The number of objects the Space Station encounters is a different calculation because it involves a shorter time period of measurement and a different SOI cross-sectional area. Since I am ultimately interested in the impact of avoidance maneuvers on the Space Station fuel supply, and since the Space Shuttle will resupply the Space Station every 90 days, I must calculate the number of encounters in each 90-day period. Furthermore, avoidance maneuvers will be required not only if the object lies directly in the Space Station's path, but also if it is in close proximity to a collision path. The reason for this lies in the present inaccuracies of the ground-based

tracking facilities, on which the Space Station will depend for advance warning. These inaccuracies are such that there may be up to ten kilometers of error in the tracking of any satellite (35:21). Therefore, a buffer or safety zone surrounding the Space Station must be created so that avoidance maneuvers are required should a debris object enter this zone.

The actual calculation of the number of encounters follows that of Hecht, as discussed in chapter two. The calculation is also similar to that made in determining λ, since both concern themselves with the number of objects found within the volume swept out by the Space Station as it travels through space. I calculate the volume swept out by the Space Station for one revolution about the earth in the initialization subroutine, which is labeled SOIVSM and is of the form:

$$SOIVSM = 2\pi aA$$

where

a = SOI altitude from earth's center (km)

A = buffer zone area (km^2)

Based upon a conversation with Mr. Redding from Johnson Space Center, I initially chose a ten kilometer diameter circular buffer zone with the Space Station at the center (28). At the planned orbital altitude of 500 kilometers, the value of SOIVSM becomes:

$$SOIVSM = 2\pi \times (6378km + 500\ km) \times 2\pi \times (10km)^2$$

$$= 2\pi \times (6878 \text{ km}) (2\pi \times 100 \text{ km}^2)$$

$$= 13{,}576{,}628 \text{ km}^3/\text{orbit}$$

I next need to calculate the circular orbital period of the Space Station in order to ultimately determine the number of orbits made by the spacecraft in one week, which is the time period of interest. The circular orbital period is:

$$T_C = 2\pi \, (r^3/\mu)^{1/2} \qquad (7:366)$$

where

r = orbital radius from earth's center (km)

μ = universal gravitational constant (km^3/sec^2)

The actual calculation becomes:

$$T_C = 2\pi \, [(6878)^3/398{,}601.3]^{1/2}$$

$$= 94.613372 \text{ min/orbit}$$

The orbits swept out per unit time are:

94.613372 min x 1 hr/60 min = 1.5768895 hr/orbit

1 orbit/1.5768895 hr x 24 hr/1 day = 15.219836 orbits/day

15.219836 orbits/day x 7 days/1 wk = 106.53885 orbits/wk

Multiplying SOIVSM by the number of orbits made per week yields the volume swept out by the Space Station buffer area per week, as follows:

$$13{,}576{,}628 \text{ km}^3/\text{orbit} \times 106.53885 \text{ orbits/wk}$$

$$= 1.4464383 \times 10^9 \text{ km}^3/\text{wk}$$

This number can now be multiplied by the debris spatial

density (RHOMD) to obtain the number of objects encountered per week. These calculations, summed over 13 week periods, yield the number of encounters over each 90-day period requiring avoidance maneuvers.

Inter-Object Collision Subroutines

The inter-object collision event consists of periodically checking whether an inter-object collision occurred in any of the altitude bands. This event is divided into three separate subroutines (IOCOLL, IOCOLM, IOCOLH) corresponding to the altitude band in which the collision occurs. This was done to increase the readability of the computer coding involved with this event. Except for the difference in several values used by the subroutines, their structure is virtually identical as depicted in Figures 4.11, 4.12, and 4.13. The following discussion covers the basic functions these subroutines share as well as highlighting the differences between them.

The first function the subroutines perform is to schedule the next check to see if a collision occurred. I initially selected the time in between checks to be one day. However, I realized that if results showed one collision constantly occurring each day, the debris spatial density could be large enough to support more than one collision per day. Therefore, I would have to decrease the time in between checks.

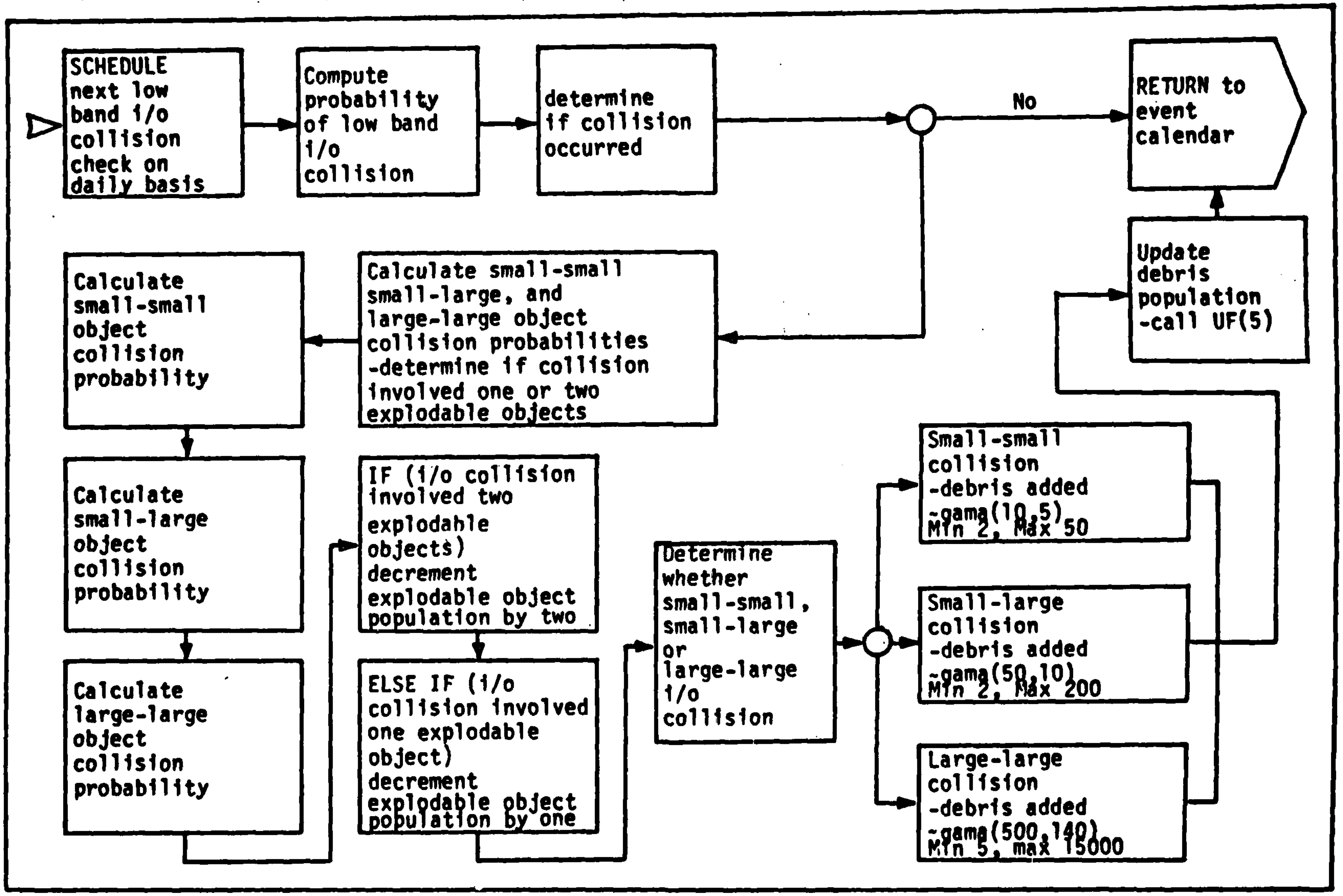


Figure 4.11. Low Altitude Band Inter-Object Collision (IOCOLL) Subroutine Flow Diagram

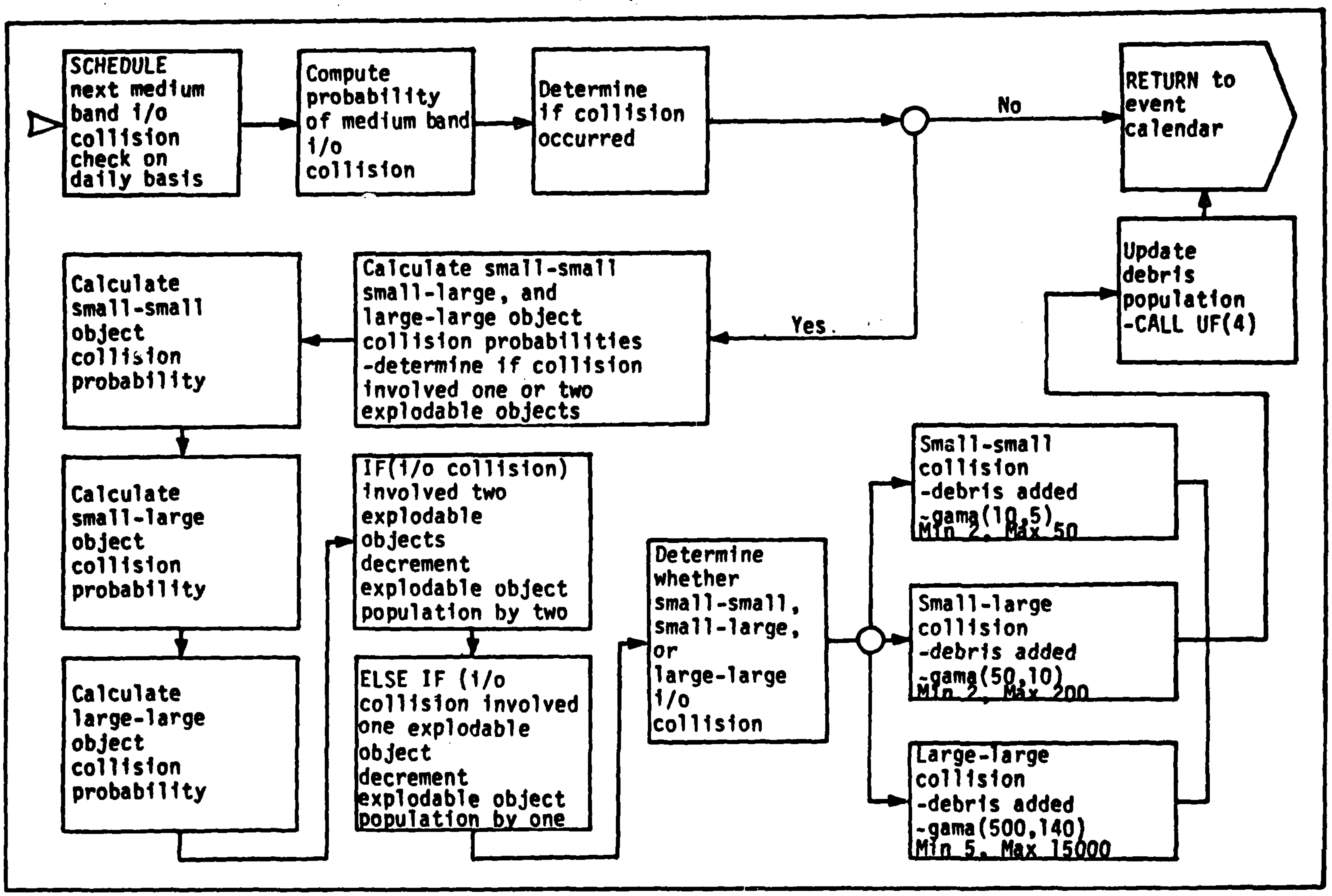


Figure 4.12. Medium Altitude Band Inter-Object Collision (IOCOLM) Subroutine Flow Diagram

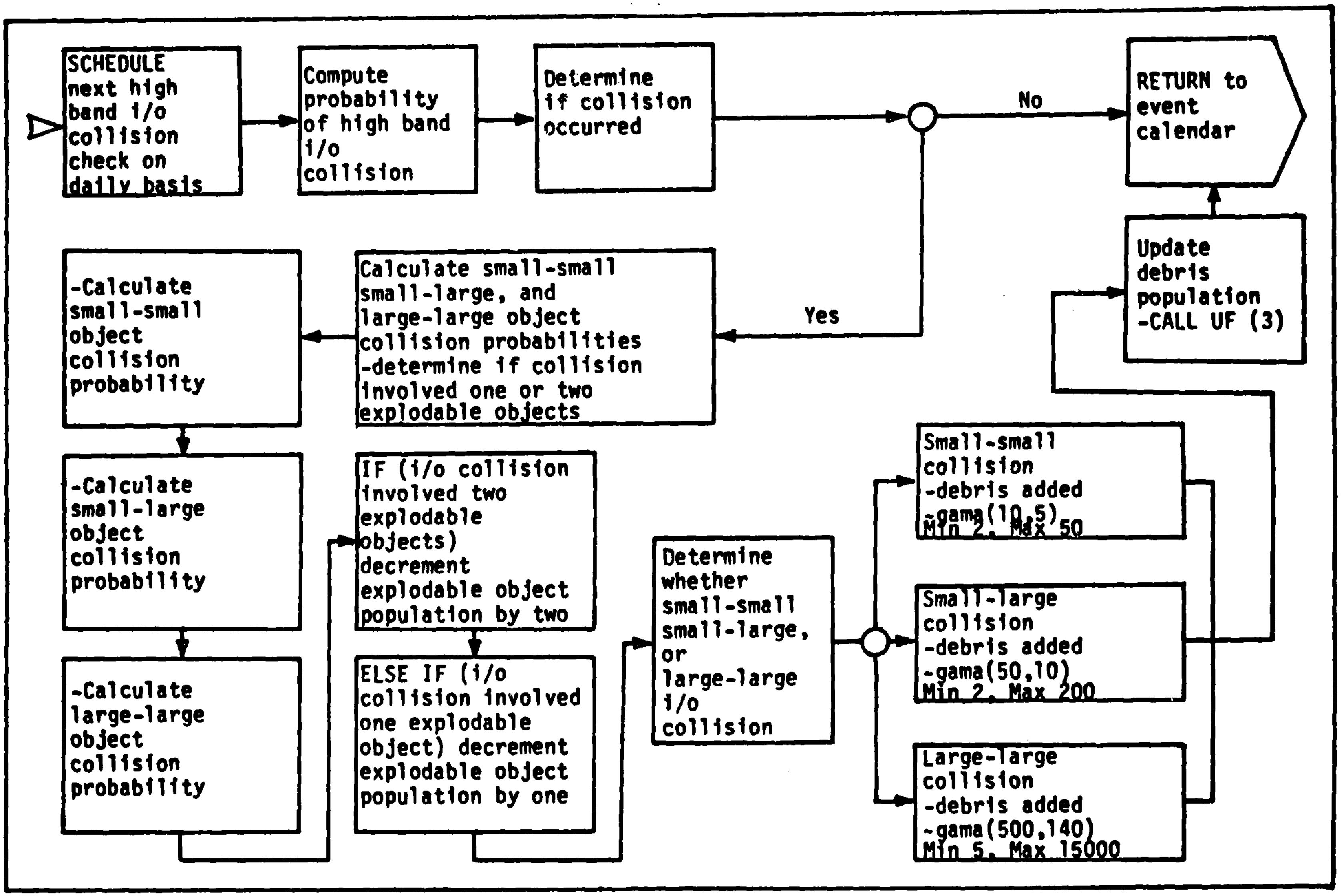


Figure 4.13. High Altitude Band Inter-Object Collision (IOCOLH) Subroutine Flow Diagram

The next step is to calculate the probability of collision between any two objects other than the Space Station for each altitude band using parameter values found at that instant in time. The form of the probability calculation is primarily the same as that for calculating the Space Station collision probability as presented earlier in this chapter. For example, the inter-object collision probability calculation for the high altitude band is:

COLIDE = [1 - exp(-satphi x c)] x (satphi/2)

where

c = (1/volume of high altitude band) x average object area (high band) x average relative velocity (high band) x time period

$c = 1/\{4/3 \times [(7278\text{ km})^3 - (6978\text{ km})^3] \times 0.0000013\text{ km}^2 \times (0.6 \times 7.478829\text{ km/sec}) \times 60{,}4800\text{ sec}\}$

$= 1.8416597 \times 10^{-11}$

Multiplication of the collision probability calculation by half of the debris population in a particular altitude band generates the overall probability of any two objects colliding, yet avoids the "double counting" of probabilities. For example, if there were two objects in a particular altitude band, the probability that the two objects would collide would be:

P(object #1 colliding with object #2)

and not

P(object #1 colliding with object #2)

+P(object #2 colliding with object #1)

since I am adding, or "double counting", the same probability twice.

The inter-object collision subroutines next determine whether a collision actually occurred. This is accomplished by generating a random variate from a uniform (0,1) distribution and determiningif that value is less than the previously calculated collision probability. If it is not, SLAM exits the subroutine and returns to the event calendar. If a collision does occur, the subroutine increments the collision rate for that year in the appropriate altitude band and calculates the probability that the collision was between two small objects, a small and a large object, or two large objects. It also calculates whether the collision involved either one or two explodable objects. These probability calculations involve the use of combinations. For example, if there are eight small objects and two large objects in a particular altitude band, the probability of two small objects colliding involves choosing any two objects from the small object population divided by the choosing of two objects out of the total population. In equation form:

$$\binom{8}{2}/\binom{10}{2} = (8!/2!6!)/(10!/2!8!)$$
$$= [(8 \times 7)/2]/[(10 \times 9)/2]$$
$$= 56/90$$

The probability that the collision involves a small and a large object requires that one object be chosen out of both the small and large object populations. In equation form:

$$\binom{8}{1}\binom{2}{1}/\binom{10}{2} = (8!/1!7!)(2!/1!1!)/(10!/2!8!)$$
$$= [(8/1)(2)/(10x9)/2] = 2(16/90)$$
$$= 32/90$$

Finally, the probability that two large objects collide is:

$$\binom{2}{2}/\binom{10}{2} = (2!/2!0!)/(10!/2!8!)$$
$$= 1/[(10 \times 9)/2]$$
$$= 2/90$$

Note that these probabilities sum to equal one. The method used to calculate the probability that the collision involved one or two explodable objects is identical to that used above, except that the population in each altitude band is divided into explodable and non-explodable objects. Since the probability that two explodable objects would be involved in the collision is smaller than the probability involving one explodable object, the random variate is checked with the smaller probability initially. If the random variate is less than this probability, I decrement the explodable object

population in the altitude band where the explosion occurred by two. If it is greater, I check to see if it is still less than the one explodable object collision probability. I decrease the explodable object population by one if it is.

The next step involves using the previously calculated collision probabilities for the various object sizes to determine what type of collision actually occurred. Another random variate is generated and compared to these probabilities. Once the type of collision is determined, I calculate the amount of debris generated from the collision. Both the type of distribution and the parameters used to generate this quantity came from responses to the Penny and Jones survey (24:104-105). I use the gamma distribution for each type of collision, but use different parameters for each type based upon the assumption that collisions involving larger objects will produce more particles.

The inter-object collision subroutines perform the final function of updating the debris populations in the low, medium, and high altitude bands by calling UF(3), UF(4), and UF(5), respectively. The number of small and large objects generated differ based on the type of collision. For a collision between a large and a small object, I assume that 90% of the objects

generated are small. For a large object-large object collision, I assume that 80% of the objects generated are small. These assumptions are simply estimates that match the assumptions made by Penny and Jones since I could not find any documentation giving more concrete values.

CHECK Subroutine

The purpose of the CHECK subroutine, whose flow diagram is in Figure 4.14, is to accumulate certain data points over the predetermined number of simulation runs. This subroutine is scheduled at the end of each year, and has two major functions to perform. First, it computes the average debris and explodable object populations by altitude band and in total, as well as the average Space Station collision probabilities per year. Since the orbital decay and collision probability subroutines obtained values for these variables on a weekly basis, the CHECK subroutine simply divides the summations over the year by 52 to obtain the average value. This subroutine also computes the total launch, ASAT test, unintentional explosion, and inter-object collision rates for each year by summing the rates obtained for that same year from each altitude band. The CHECK subroutine's second major function is to determine at the end of each year the 90-day period in which the Space Station

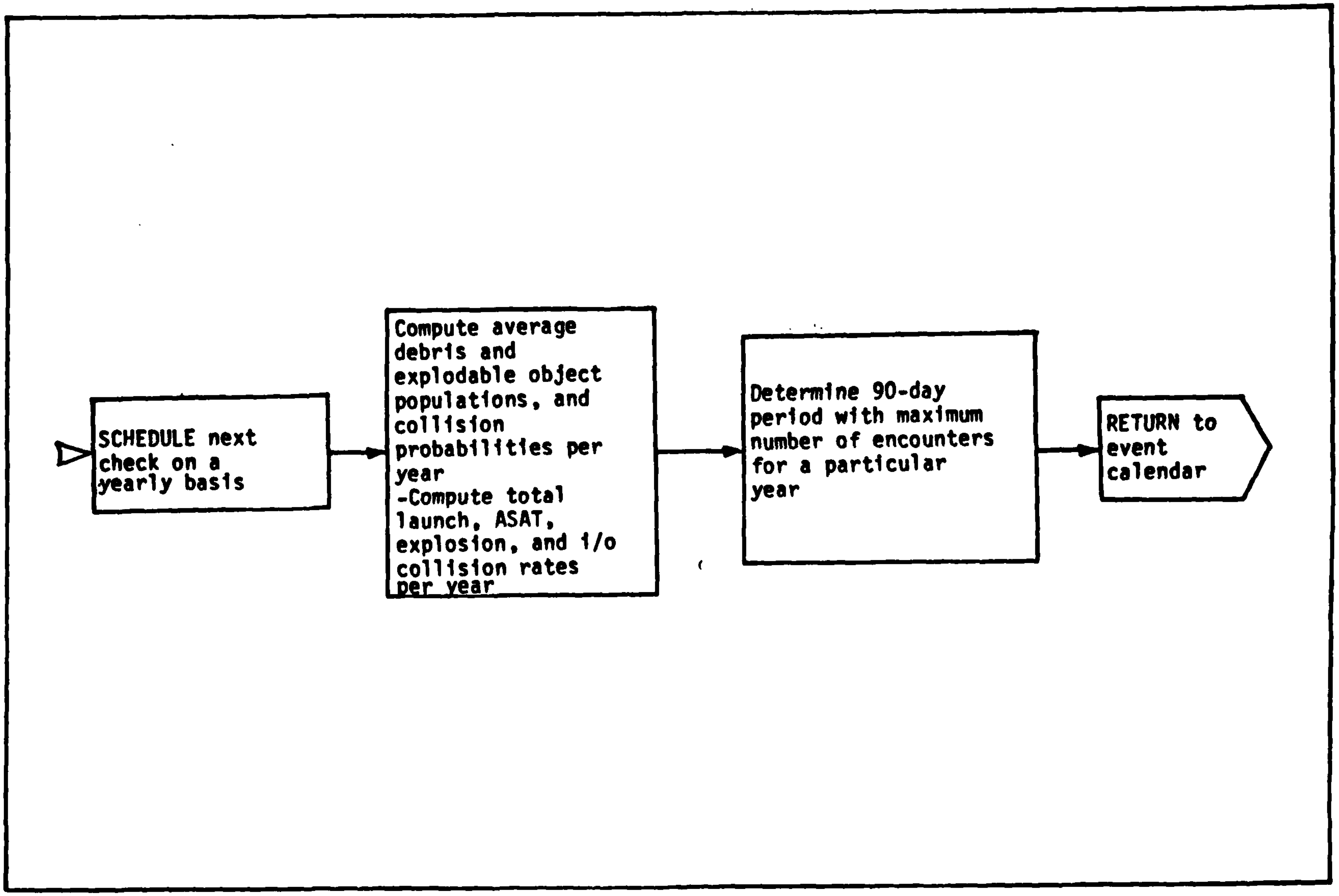


Figure 4.14. System Parameter Check (CHECK) Subroutine Flow Diagram

experienced the most encounters with debris. This value is then compared with the maximum value found for that same year in previous simulation runs in order to obtain the maximum value over all runs.

Output Subroutine

The subroutine entitled OTPUT is called by SLAM after each simulation run. Its purpose is to sum certain system parameters over the desired number of simulation runs, to compute averages of certain parameters upon completion of the runs, and to present the desired results. These functions are shown in the flow diagram presented in Figure 4.15, and an example of the output generated is presented in Appendix C.

Each function is actually performed in preparation for the accomplishment of the next function. System parameters summed by year over the number of simulation runs include debris population, explodable object population, launch rate, ASAT test rate, unintentional explosion rate, inter-object collision rate, the number of encounters for each altitude band, and the SOI probability of collison. Upon completion of the simulation runs, the subroutine uses the above summations to calculate averages of the variable listed above by year. The actual presentation of output includes, by altitude band and year, the average probability of collision, its variance, the average

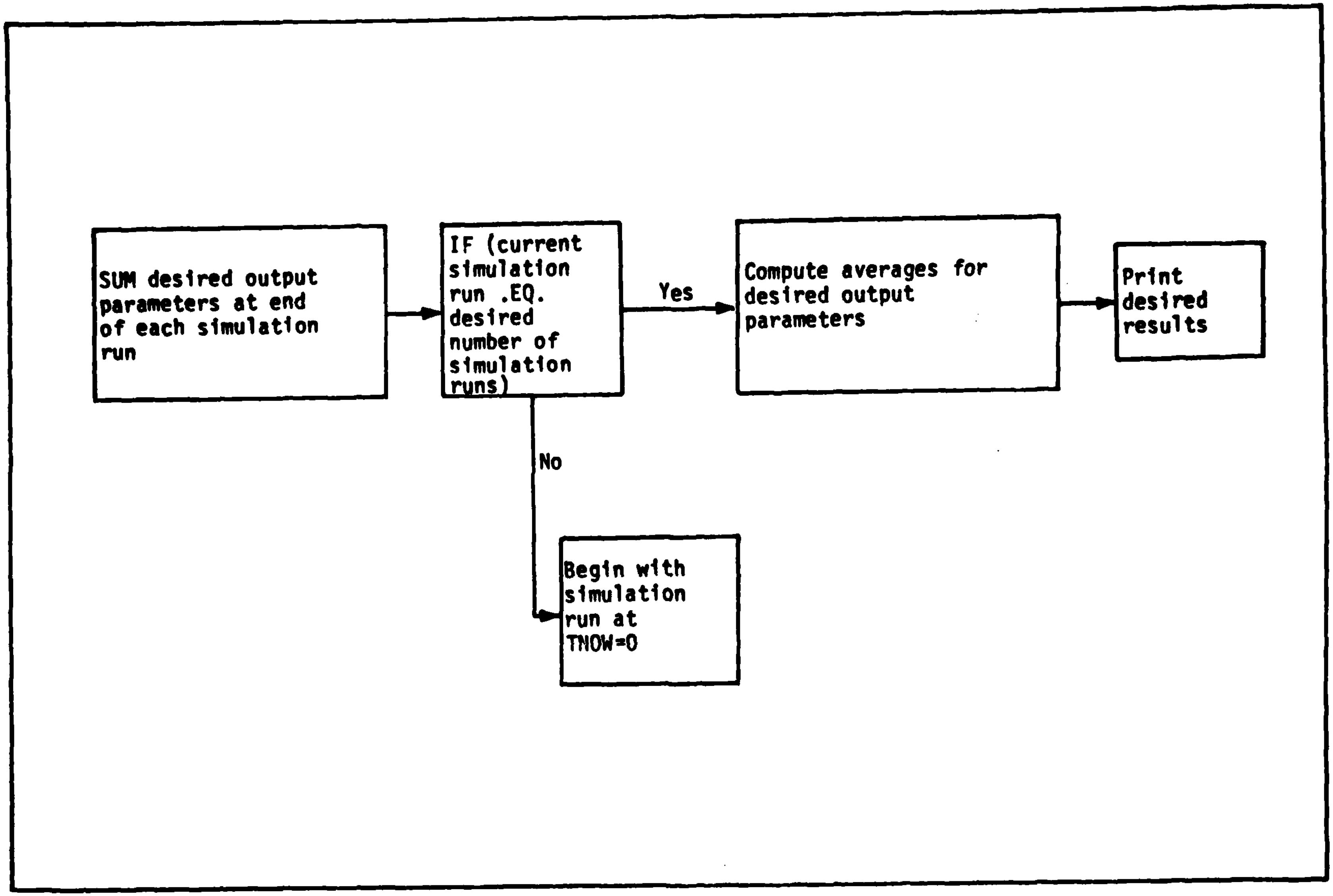


Figure 4.15. Output (OTPUT) Subroutine Flow Diagram

number of encounters, the maximum number of encounters over a 90-day period for that year, the average number of launches and its percentage of the total, the average number of explosions and inter-object collisions, the average debris population and its percentage of the total, and the average expodable object population and its percentage of the total. I also include as output, by year, the total debris and explodable object populations as well as the total launch, explosion, ASAT test, and inter-object collision rates.

Conclusion

I designed the space debris environment model to a level of complexity sufficient to enable me to obtain collision probability and debris encounter calculations for the Space Station through the year 2013. This level of complexity, combined with the use of discrete-event simulation, let me observe the dynamics of the system and analyze the sensitivity of the above calculations to changes in model parameters. The next two steps in the model development process, verification and validation, enabled me to perform this analysis with confidence.

V. Model Verification and Validation

Introduction

Together, verification and validation act as the "bridge" between parametric model development and confident analysis of resulting model output. Verification consists of comparing the conceptual model with the parametric model to determine if elements within the conceptual model are correctly implemented in the computer code. Validation involves the determination as to whether the parametric model is an accurate representation of the real system. Verification is usually a straight-forward process. However, validation depends on how much is known about the real system. Based upon the uncertainties present concerning the space debris environment system as discussed in earlier chapters, validation was much more difficult to accomplish than verification. The following sections cover the efforts made to verify and validate the space debris environment parametric model.

Verification

Verification, according to Banks and Carson, asks two questions concerning the conceptual model:

> "Is the model implemented correctly in computer code? Are the input parameters and logical structure of the model correctly represented in the code?" (1:376-377)

The answer to the first question is "yes". The discrete event simulation directly follows the organization of the conceptual model as represented by the causal diagram (Figure 3.1). Each system element is included in the computer model as a separate event subroutine. The second question is a little more difficult to answer, requiring an examination of the inner workings of the parametric model. In order to analyze the proper operation of the parametric model, I incorporated extensive verification statements into the computer code during the initial simulation runs. These statements were simply Fortran "print" statements developed for each variable potentially changing in value within a subroutine. I added verification statements at the beginning and end of each subroutine in order to track the change in any variables. This technique, combined with the "trace" feature of the SLAM simulation language which itself keeps track of changing attribute values, allowed for the analysis of changing system parameters over the simulation run time. In addition, statements were added at the beginning of each subroutine and user function to verify the correct scheduling of the event on the event calendar, the actual occurrence of that event at the appropriate time, and the calling of the appropriate user function.

I ran the trace for one simulation run of the baseline model. The baseline consisted of a starting untracked object population three times the tracked populations and a 10 kilometer Space Station encounter buffer zone. After simulating the entire 30 years of interest, I checked the change in various parameter to beginning of quote magnitudes over the entire period. Overall, the parametric model, as designed, schedules events correctly, passes the appropriate values between subroutines, and makes the appropriate calculations. The second question can therefore be answered affirmatively, and comparison of the parametric model to the real system can proceed.

Validation

According to Banks and Carson, validation is required for two reasons:

> "(1) to produce a model that represents true system behavior closely enough for the model to be used as a substitute for the actual system for the purpose of experimenting with the system; (2) to increase to an acceptable level the credibility of the model, so that the model will be used by managers and other decision makers" (1.376).

The difficulty in achieving the first goal is that, as chapter two indicated, true system behavior is not known. The magnitude of the untracked debris population, the actual number of potentially explodable

objects, and the accurate identification of explosion and inter-object occurrences and their dynamics are but a few of the critical unknowns. Therefore, a validation depended on a comparison of the parametric model to that space debris environment behavior which is known, to interactions predicted in the conceptual model, and to the array of the past models which have been of use to other researchers. The validation process itself involved the determination of whether the parametric model had a high face validity, the validation of model assumptions, and the comparison of model input-output transformations to corresponding input-output transformations for the real system (1:385). I hoped that the validation process steps combined with the areas of comparison would make the validation goals attainable. Determination of the level of face validity possessed by the space debris environment parametric model involved the decision by two individuals familiar with the space debris environment as to the reasonableness of the model. The two individuals were Lieutenant Colonel Mark Mekaru (USAF), who was involved with the Penny and Jones effort, and Dr. Wiesel, who has conducted research concerning the space debris environment. Both men agreed that the model was reasonable in describing the environment at a level commensurate to the purpose for

which it was intended.

The validation of model assumptions entailed comparing model output with space debris environment behavior which is known and with behavior predicted by the conceptual model. The only system behavior known with any certainty are past and present space launch and ASAT test rates. Running nine iterations of the baseline model, whose output is presented in Tables 5.1, 5.2, 5.3, and 5.4, provided the necessary data to compare with these historical rates. Recall that Kessler predicted the launch rate into the altitude bands of interest to be between 120 and 150 launches per year, and that recent data from the TRW Space Log supported this prediction. The model produced, on the average, 137.4 launches per year with a standard deviation of 1.43. This is well within the predicted values. Furthermore, analysis of recent data from the TRW Space Log indicated that, historically, 69% of the launches entered the low altitude band, 15% entered the medium band, and 16% entered the high band. Actual model output indicated that, on the average, 68.63% of the launches entered the low altitude band with a standard deviation of 2.32%. Again, this is within a standard deviation of the historical value. Likewise, the model saw 14.4% of the launches enter the medium band with a standard deviation of 2.78%, and 15.45% of

Table 5.1

Low Altitude Band Baseline (3x10km) Model Output

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	96	.7059	.22	.00	.00
1985	94	.6912	.44	.11	.67
1986	93	.6838	.44	.56	.56
1987	91	.6691	.67	.44	.89
1988	102	.7234	.56	.44	.33
1989	92	.6765	.56	.11	.89
1990	90	.6522	.89	1.00	1.00
1991	92	.6765	1.11	1.11	.56
1992	98	.7050	1.22	1.89	.44
1993	96	.7111	.56	2.22	1.22
1994	99	.7226	.89	.78	.89
1995	95	.6934	.33	1.78	.33
1996	91	.6547	1.22	3.44	.78
1997	97	.6978	.78	3.11	.56
1998	95	.6884	.56	1.78	.67
1999	100	.7246	.78	.89	1.11
2000	91	.6500	.78	1.56	.33
2001	94	.6861	1.11	1.89	.44
2002	88	.6377	.67	1.11	.33
2003	97	.7080	.67	2.44	1.00
2004	94	.6812	.56	.67	.67
2005	94	.6812	.89	2.00	.22
2006	93	.6889	.22	.67	.33
2007	92	.6715	.22	.89	.56
2008	98	.7101	.44	1.11	.22
2009	90	.6522	.78	.89	.22
2010	93	.6691	.89	.67	.56
2011	92	.6715	.78	1.00	.67
2012	96	.7059	.67	.44	.22
2013	96	.7007	1.11	1.11	.44

year	avg debris pop size	% total	avg exp pop size	% total
1984	2472	.1137	164	.2108
1985	8915	.2126	275	.3183
1986	14477	.2021	339	.3717
1987	19123	.1977	372	.4017
1988	26782	.2336	405	.4322
1989	28407	.2213	427	.4567
1990	32277	.2282	427	.4641
1991	39402	.2440	425	.4691
1992	43870	.2553	425	.4830
1993	43308	.2351	418	.4912
1994	40895	.2124	415	.5086
1995	40606	.2012	406	.5192
1996	41076	.1954	398	.5244
1997	41535	.1933	393	.5311
1998	38225	.1761	387	.5443
1999	36302	.1643	383	.5599
2000	35013	.1515	376	.5688
2001	39194	.1599	371	.5797
2002	41265	.1600	366	.5856
2003	36798	.1414	363	.5970
2004	35295	.1331	361	.6088
2005	37341	.1345	359	.6222
2006	30736	.1061	355	.6317
2007	26395	.0895	351	.6405
2008	23301	.0782	353	.6574
2009	25636	.0810	351	.6698
2010	30792	.0941	352	.6795
2011	31852	.0958	352	.6916
2012	30025	.0887	352	.7111
2013	34047	.0947	351	.7282

Table 5.2
Medium Altitude Band Baseline (3x10km) Model Output

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	20	.1471	.33	.00	.22
1985	18	.1324	.11	.00	.22
1986	22	.1618	1.00	.22	1.11
1987	22	.1618	.56	.33	.56
1988	18	.1277	.11	.78	.44
1989	21	.1544	.44	.33	1.22
1990	23	.1667	.11	.78	.44
1991	22	.1618	.22	.22	.89
1992	19	.1367	.33	.44	.33
1993	20	.1481	.33	.33	1.00
1994	20	.1460	.11	.33	1.56
1995	19	.1387	.11	1.00	.89
1996	22	.1583	.11	.67	.33
1997	18	.1295	.11	.44	.22
1998	20	.1449	.00	.78	1.67
1999	20	.1449	.11	.22	1.56
2000	22	.1571	.22	.44	.89
2001	21	.1533	.22	.44	.44
2002	22	.1594	.00	.22	1.44
2003	17	.1241	.00	.78	1.33
2004	23	.1667	.00	.22	1.11
2005	19	.1377	.00	.00	1.33
2006	17	.1259	.00	.44	.67
2007	21	.1533	.00	.22	1.00
2008	23	.1667	.00	.78	1.22
2009	24	.1739	.00	.44	.33
2010	22	.1583	.00	.00	1.11
2011	23	.1679	.00	.00	1.67
2012	18	.1324	.00	.00	.44
2013	20	.1460	.11	.00	1.22

year	avg debris pop size	% total	avg exp pop size	% total
1984	5599	.2576	220	.2828
1985	8217	.1960	206	.2384
1986	15134	.2113	197	.2160
1987	22087	.2283	190	.2052
1988	21928	.1913	177	.1889
1989	23238	.1810	166	.1775
1990	22006	.1556	156	.1696
1991	20670	.1280	152	.1678
1992	20722	.1206	139	.1580
1993	23029	.1250	129	.1516
1994	20937	.1087	117	.1434
1995	20329	.1007	104	.1330
1996	18911	.0900	95	.1252
1997	17029	.0793	84	.1135
1998	15727	.0725	69	.0970
1999	14430	.0653	59	.0863
2000	16320	.0706	51	.0772
2001	15978	.0652	44	.0688
2002	16045	.0622	38	.0608
2003	14360	.0552	28	.0461
2004	13082	.0493	23	.0388
2005	12061	.0434	19	.0329
2006	11328	.0391	12	.0214
2007	10716	.0364	8	.0146
2008	10413	.0350	8	.0149
2009	10155	.0321	11	.0210
2010	9947	.0304	10	.0193
2011	10001	.0301	9	.0177
2012	9983	.0295	7	.0141
2013	11373	.0316	4	.0083

Table 5.3
High Altitude Band Baseline (3x10km) Model Output

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	19	.1397	.89	.22	.78
1985	23	.1691	.78	.22	.11
1986	20	.1471	1.22	1.11	.33
1987	21	.1544	1.00	1.56	.56
1988	20	.1418	.44	1.67	.22
1989	21	.1544	.78	2.78	.89
1990	23	.1667	.78	3.78	.56
1991	21	.1544	.22	5.44	.56
1992	21	.1511	.56	6.22	.22
1993	18	.1333	.56	6.56	.78
1994	16	.1168	1.11	8.67	.56
1995	22	.1606	.33	8.11	.78
1996	25	.1799	.33	14.67	.89
1997	23	.1655	.11	15.33	.22
1998	22	.1594	.78	14.78	.67
1999	17	.1232	.33	17.89	.33
2000	25	.1786	.67	16.56	.78
2001	20	.1460	.56	21.44	.11
2002	26	.1884	.33	25.89	.22
2003	22	.1606	.44	30.22	.67
2004	20	.1449	.22	25.56	.22
2005	24	.1739	1.00	34.11	.44
2006	23	.1704	.89	33.44	.00
2007	23	.1679	.11	36.33	1.44
2008	16	.1159	.33	40.89	.56
2009	22	.1594	.22	48.78	.44
2010	23	.1655	.22	47.00	.33
2011	20	.1460	.00	48.44	.67
2012	21	.1544	.11	51.33	.33
2013	20	.1460	.22	51.78	.33

year	avg debris pop size	% total	avg exp pop size	% total
1984	13660	.6285	393	.5051
1985	24793	.5913	382	.4421
1986	42010	.5866	375	.4112
1987	55536	.5740	363	.3920
1988	65929	.5751	353	.3767
1989	76734	.5977	340	.3636
1990	87181	.6163	335	.3641
1991	101411	.6280	327	.3609
1992	107274	.6242	314	.3568
1993	117903	.6399	302	.3549
1994	130695	.6788	282	.3456
1995	140928	.6981	270	.3453
1996	150217	.7146	265	.3491
1997	156268	.7274	261	.3527
1998	163104	.7514	253	.3550
1999	170167	.7703	240	.3509
2000	179834	.7779	232	.3510
2001	189953	.7749	223	.3484
2002	200516	.7777	220	.3520
2003	209165	.8035	216	.3553
2004	216747	.8175	207	.3491
2005	228300	.8221	197	.3414
2006	247503	.8547	193	.3434
2007	257657	.8741	187	.3412
2008	264182	.8868	174	.3240
2009	280728	.8869	159	.3034
2010	286341	.8754	154	.2973
2011	290636	.8741	146	.2868
2012	298608	.8818	135	.2727
2013	313952	.8736	125	.2593

Table 5.4
Baseline (3x10km) Model Summary Results

year	tot pop	tot explod pop
1984	21733	778
1985	41927	864
1986	71622	912
1987	96748	926
1988	114641	937
1989	128381	935
1990	141466	920
1991	161485	906
1992	171868	880
1993	184242	851
1994	192529	816
1995	201865	782
1996	210206	759
1997	214833	740
1998	217057	711
1999	220900	684
2000	231168	661
2001	245127	640
2002	257827	625
2003	260326	608
2004	265125	593
2005	277704	577
2006	289569	562
2007	294770	548
2008	297898	537
2009	316521	524
2010	327082	518
2011	332490	509
2012	338618	495
2013	359375	482

year	# launches	# explosions	# asat tests	#i/o coll
1984	136	1.44	1.00	.22
1985	136	1.33	1.00	.33
1986	136	2.67	2.00	1.89
1987	136	2.22	2.00	2.33
1988	141	1.11	1.00	2.89
1989	136	1.78	3.00	3.22
1990	138	1.78	2.00	5.56
1991	136	1.56	2.00	6.78
1992	139	2.11	1.00	8.56
1993	135	1.44	3.00	9.11
1994	137	2.11	3.00	9.78
1995	137	.78	2.00	10.89
1996	139	1.67	2.00	18.78
1997	139	1.00	1.00	18.89
1998	138	1.33	3.00	17.33
1999	138	1.22	3.00	19.00
2000	140	1.67	2.00	18.56
2001	137	1.89	1.00	23.78
2002	138	1.00	2.00	27.22
2003	137	1.11	3.00	33.44
2004	138	.78	2.00	26.44
2005	138	1.89	2.00	36.11
2006	135	1.11	1.00	34.56
2007	137	.33	3.00	37.44
2008	138	.78	2.00	42.78
2009	138	1.00	1.00	50.11
2010	139	1.11	2.00	47.67
2011	137	.78	3.00	49.44
2012	136	.78	1.00	51.78
2013	137	1.44	2.00	52.89

the launches enter the high band with a standard deviation of 17.6%. Overall, the close parallel between the model output and historical data regarding the space launch rate increased my confidence in the model for at least representing one system element accurately.

The second aspect of system behavior that could be readily compared to model output involved historical ASAT test rates. Based upon past Soviet ASAT test occurrences, the space debris environment model employed an average of two ASAT test occurrences per year by all nations combined, with no less than one or no more than three occurring in a year over the entire simulation run. In addition, the model accounted for twice as many ASAT tests occurring in the medium altitude band as in the other bands per year based on the historical Soviet ASAT test data. Model results yielded values very close to the historical target values. The mean total ASAT test rate over the entire model run time was equal to 1.967 with a standard deviation of 0.7649. This value closely approximates the historical test rate supported by the literature. The mean ASAT test rates per year in the low and high altitude bands over nine iterations of the model were 0.5703 and 0.50, respectively, with accompanying standard deviations of 0.3011 and 0.305. The mean rate

within the medium altitude band was 0.8953 with a standard deviation of 0.4754. This translates to the low altitude band ASAT test rate being 63.69% of the medium band rate while the high altitude band ASAT test rate is 55.85% of the medium band rate. While the medium band rate is not exactly twice the rate of the other altitude bands, it is relatively close. The large standard deviations indicate the randomness which characterizes these tests, and in fact may more accurately represent the variability in the number of future tests conducted by any number of nations.

Many elements of the space debris environment system could not be used directly to validate the simulation model because of a lack of historical data. Therefore, I had to rely on the causal diagram representing the conceptual model to compare predicted trends with trends identified in the simulation output. The trends I investigated consisted of the relationship between explosions and the debris population, ASAT tests and the debris population, inter-object collisions and the debris population, and the predicted self-sustaining nature of inter-object collisions. The following paragraphs discuss the comparison of these relationships with the simulation model results.

The conceptual model predicts that increases in

unintentional explosions, ASAT tests, and inter-object collisions each have a corresponding positive effect on the space debris population. Although the occurrences of these events are distinct and their effects separate, the model output does not provide for that distinction. Therefore, validation of the above conceptual model trends using parametric model results is not entirely straightforward. However, certain data points do contribute to the validation process. In Table 5.1, for example, the average of 0.22 explosions in the year 1984 are the only major sources of debris occurring in that year. As expected, an increase of the starting low altitude band population from 1,244 to an average of 2,472 occurs. Table 5.2 provides additional evidence concerning the positive effect explosions, ASAT tests, and inter-object collisions have on the debris population. Substantial increases in the occurrence of the three sources of debris from 1985 to 1986 results in almost a 100% increase in the average debris populations for those years. Table 5.3 shows a very similar occurrence for those same years in the high altitude. Overall, the above examples indicate that the parametric model does indeed adequately represent the relationships between the major sources of debris and the debris population as logically predicted in the conceptual model.

The only positive loop in the causal diagram that is part of those system elements included in the model involves inter-object collisions and the debris population. The loop represents the prediction that inter-object collisions will increase the debris population which in turn will increase the number of collisions, forming a self-sustaining relationship. The high altitude band depicts this predicted phenomena because its debris population constantly increases due to an insignificant decay rate and a large number of unintentional explosions. As Table 5.3 indicates, the inter-object collision rate continues to grow along with the debris population over the years.

The comparison of model input-output transformations to corresponding input-output transformations for the real system is limited because actual data on Space Station collision rates does not exist. However, other model and study results exist from which my results can be checked for reasonableness. Herbert Hecht calculated in 1970 that a space station 100 meters in diameter at an altitude of 500 kilometers would have a collision rate of 0.005 per year (9). This translates to at least one collision occurring within 200 years. The space debris environment model, on the other hand, calculates the maximum collision probability to be 0.0494852 in the

Table 5.5
Baseline (3x10km) Model Collision Probabilities

year	probability collision	variance	# encounters	max enc per qtr
1992	.04417984	.00019094	13073.00	5048
1993	.04948522	.00032979	14529.56	6902
1994	.04559287	.00033107	13208.33	5515
1995	.04469331	.00051639	12825.00	6325
1996	.04213160	.00040907	11928.56	5710
1997	.03847091	.00027838	10738.00	4681
1998	.03599253	.00021602	9914.78	3919
1999	.03342433	.00019064	9095.78	5082
2000	.03807677	.00028706	10289.78	4873
2001	.03743685	.00037330	10074.22	5905
2002	.03762065	.00034473	10116.89	5359
2003	.03377774	.00022712	9050.89	4498
2004	.03084769	.00014411	8242.89	3784
2005	.02849567	.00009268	7598.67	3202
2006	.02679938	.00006231	7134.33	2764
2007	.02537565	.00004287	6749.22	2434
2008	.02467216	.00003342	6554.89	2338
2009	.02407048	.00002881	6394.67	2241
2010	.02358310	.00002774	6260.00	2169
2011	.02370759	.00003019	6296.22	2128
2012	.02366490	.00003568	6284.56	2221
2013	.02687203	.00012504	7163.22	3666

year 1992 with an untracked debris population three times that of the tracked population. Table 5.5 presents the calculated collision probabilities. The maximum value translates to at least one collision occurring within only 20 years.

The fact that this collision rate is exactly 10 times larger 22 years after Hecht's calculation can be explained in two ways. First, Hecht obviously used a debris population commensurate to that time period. As other literature discussed in chapter two indicates, the debris population has continued to grow ever since Sputnik, so an increase in the collision rate over the

22 years since Hecht's calculation can be expected. Second, Hecht did not consider untrackable debris in his calculation. The consideration of this parameter in my calculations would naturally increase the collision rate depending on the estimation of the untracked debris population. This combination of an increasing trend in the debris population growth as well as the consideration of untrackable debris makes the possibility that a 10 fold increase in the collision rate can occur over 22 years seem feasible. Recent statistical analysis by a leading expert in this field does much in bolstering confidence in the above statement.

A recent statistical analysis by Donald Kessler added considerable credibility to my model results. Kessler's analysis of the proposed Space Station probability of collision operating in the current trackable debris environment resulted in the prediction that at least one collision would occur with certainty over a period of 100 years (29:16). This value decreases to one collision in 25 years if untrackable debris three times the trackable debris population is considered. Kessler's finding is slightly more optimistic than my worst-case collision rate of one in 20 years, but is slightly less optimistic than the collision rate averaged over all years of interest--at

least one collision in a 29 year period. Taking into account the possible difference in the parameters used in the calculations and in the calculation techniques themselves, I feel that the collision probability results obtained by the space debris environment model are reasonable.

Conclusion

The verification and validation process applied to the parametric model was not straight-forward. While verification was accomplished through extensive computer coding, validation required the use of various standards of comparison and a certain degree of abstraction. True system behavior is not known, so the conceptual model and past analyses had to be used for comparison purposes. Overall, the parametric model's operation and output appears to be a reasonable representation of the predicted environment. While more complete validation can occur once the space debris environment system is more fully understood, I feel my validation is sufficient to instill confidence in the output as being initial estimates of the severity of the collision problem.

VI. Analysis

Introduction

The potential exists within the space debris environment model to be able to analyze each element in the system and the interactions between them, as well as analyzing the resulting collision probabilities and debris encounter rates. However, I designed the parametric model to represent the system in sufficient detail to provide initial indications of any problems posed to the Space Station by the debris environment. Design of the model was also constrained by the many remaining unknowns concerning the space debris environment. Therefore, I restricted my analysis to the collision probability and debris encounter output and only included analysis of system elements as it related to explaining the above.

Of all the remaining unknowns concerning the space debris environment, the most critical unknown directly affecting the model output of interest concerns the actual size of the untracked debris population. I therefore altered the starting debris populations to account for untrackable debris populations five and eight times the amount of trackable objects. The subsequent model was compared to the baseline model which incorporated an untrackable debris population three times as large. A factor of eight was chosen to

correspond to recent observations performed with the U.S. Air Force GEODSS telescope system in estimating the amount of trackable debris (32:16). The factor of five was chosen to provide a middle range for analysis.

Recall that the calculation of the number of encounters with debris requires that a safety or buffer zone be established around the SOI. The size depends on the confidence in the exact location of the orbiting debris. To avoid the possibility of colliding with debris, the SOI would need to maneuver. A trade-off exists between the level of confidence and the fuel required for avoidance maneuvers. Therefore, I conducted analysis on the parametric model considering the number of debris encounters for one, three, five, seven, and ten kilometer radius buffer zones and the variable untrackable debris populations discussed earlier.

The remaining sections of this chapter cover the determination of the appropriate sample size needed to obtain credible results, analysis of the probability of collision calculations based on differing untrackable debris densities, and analysis of the debris encounters depending on both the buffer zone selected and the untrackable debris population. I conclude the chapter by providing sensitivity analysis results on several densities, and hence, Space Station survivability.

Sample Size Determination

The importance of determining the number of independent replications needed lies in decreasing the variance present in the output from run to run. Banks and Carson achieve this goal by determining the sample size R needed to satisfy the equation

$$R > (t_{\alpha/2,\ R-1}/\varepsilon)^2$$

where $t_{\alpha/2,R-1}$ = t - distribution

S_0 = initial estimate of population variance

ε = desired accuracy in estimating output mean (1:427)

I initially ran five independent replications of the baseline model to obtain an initial estimate of the population variance. I chose the maximum collision probability variance resulting from the five runs for this initial estimate with the belief that it would be the most restrictive estimate requiring the greatest number of runs. Using an α of 0.975 and a desired accuracy of 0.001, I calculated the minimum number of replications needed as follows:

$$R_0 = 5$$

$$S_0^2 = R_0 \hat{\sigma}^2(\hat{\rho}) = 5(.00055169)^2 = .0000015$$

$$S_0 = .0012341$$

$$(Z_{\alpha/2} S_0/\varepsilon)^2 = [(1.96)(.0012341)/(.001)]^2$$

$$= 5.7624$$

Since $t_{\alpha/2,r-1} > Z_{\alpha/2}$, where $Z_{\alpha/2}$ represents the normal distribution, the above calculation is suitable for an initial estimate of R. The final determination of R is listed below.

$$R = 6 < 9.907$$

$$R = 7 < 9.00375$$

$$R = 8 < 8.3544$$

$$R = 9 > 8.00415$$

Therefore, nine replications of the parametric model will give collision probabilities of the desired accuracy with 97.5% confidence.

Probability of Collision Analysis

The probability of collision analysis consists of comparing both the magnitudes of the probabilities among the models with varying untrackable debris populations as well as the trend of these probabilities in the future. The following paragraphs cover each of these comparisons in more detail.

Table 6.1 presents the collision probability calculations over the 22 years of analysis for the three models using different starting untracked debris populations. The calculation of Space Station collision probabilities using untracked debris populations three times the tracked populations yielded an average value of 0.0335896 with a sample standard deviation of 0.008294 over the first 22 years after

TABLE 6.1

Collision Probabilities for Models Varying Untracked Debris Populations

YEAR	UNTRACKED DEBRIS POPULATION		
	3X	5X	8X
1992	.04417984	.05004029	.06086987
1993	.04948522	.05049875	.06216294
1994	.04559287	.05495163	.05849797
1995	.04469331	.05539890	.05242787
1996	.04213160	.05388026	.05156042
1997	.03347091	.05229689	.04698096
1998	.03599253	.04581250	.04350626
1999	.03342433	.04311085	.03893839
2000	.03807677	.03845693	.03874352
2001	.03743685	.04408584	.03530629
2002	.03762065	.04284479	.03089205
2003	.03377774	.04039241	.02822994
2004	.03084769	.03629503	.03015987
2005	.02849567	.03537226	.02941474
2006	.02679938	.03297165	.02961808
2007	.02537565	.03051218	.02772455
2008	.02467216	.02876923	.02580383
2009	.02407048	.02722227	.02411659
2010	.02358310	.02597688	.02294796
2011	.02370759	.02543776	.02253946
2012	.02366490	.02484992	.02186278
2013	.02687203	.02443635	.02467285
$\bar{x}$	.0333624	.0392552	.0366808
σ	.008222	.0107317	.0132783

initial deployment of the spacecraft. This translates to the prediction that at least one collision will occur within a 29 year period. The average collision probability over the years of interest when considering an untracked debris population five times as large as the tracked debris population is 0.0392552 with a sample standard deviation of 0.0107317. This probability indicates that at least one collision will occur in 25 years, which is expected considering the increased magnitude of the untracked debris population. However, the average collision probability for an untracked debris population eight times the number of trackable objects is 0.0366808 with a standard deviation of 0.0132783, which translates to at least one collision occurring in a period of 27 years. This discrepancy apparently can be attributed to orbital decay counteracting the initial debris population containing a large number of small, untrackable objects. A more complete explanation of this discrepancy can occur by analyzing the trends existing in the collision probabilities over the 22 years of their calculation.

The general trend of the collision probability calculations is an initial increase in value within the first four years of interest, followed by a general decrease over the remaining years. Since the collision

probability as calculated depends exclusively on the changing debris spatial density, an analysis of the medium altitude band debris population over the years of interest yields the same general trend. Figure 6.1 illustrates the similarity of these trends. It should be noted that the x-axis corresponds to the year of the simulation run, with year number one corresponding to 1984, year number nine corresponding to 1992, the first year of Space Station operations, and year number 17 corresponding to Space Station maturity in the year 2000. As mentioned earlier, certain system elements are primary contributors to this model behavior. First, only 15% of the space launches enter the medium altitude band, so launch debris and, more importantly, explodable objects are not added to that band. In addition, the relatively high orbital decay rate in the medium band compared to the high altitude band decay rate depletes both the debris and explodable object populations faster than they are added. The effect of system dynamics such as the above on the collision probability calculation warrants a more exacting look at yearly probabilities.

Analysis of the maximum and minimum collision probabilities for each model of varying untrackable debris populations provides a more accurate picture of the impact caused by this parameter. The maximum

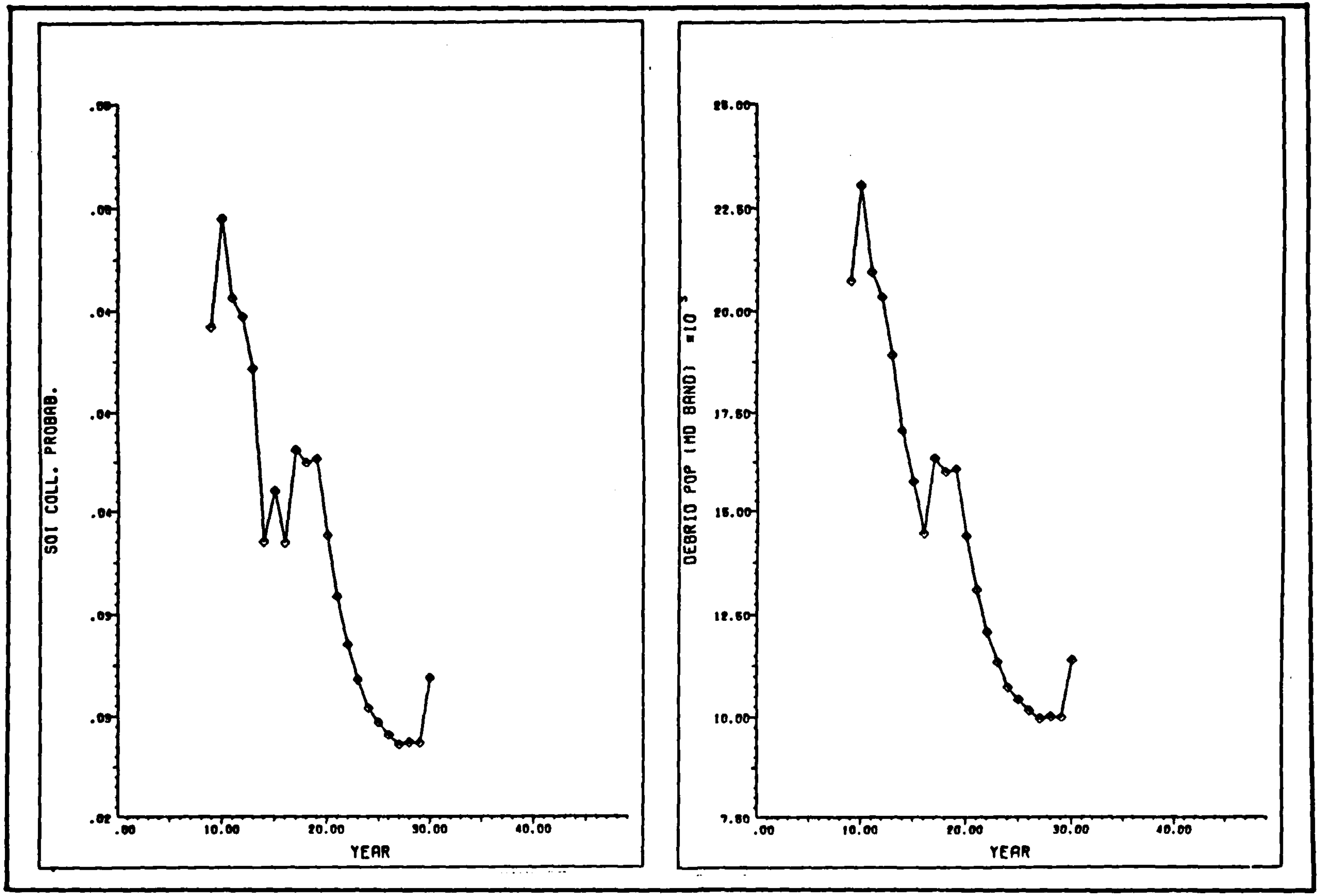


Figure 6.1· Trend of Collision Probabilities and Debris Population Over Time for 3x1km Model

probability of 0.04948522 occurs in 1993 for the model incorporating an untracked debris population three times the tracked population. The minimum value of 0.0235831 occurs in the year 2010. These probabilities translate to at least one collision occurring in 20 years and 42 years, respectively. From the model incorporating an untracked debris population five times the tracked population, 1995 yields the maximum collision probability of 0.055398, or at least one collision in only 18 years of operation. The minimum value of 0.02443635, or at least one collision in 40 years, occurs in the last year of analysis, 2013. The model using a starting tracked population eight times the tracked population results in the maximum probability occurring in 1993, with a value of 0.06216294 or at least one collision in 16 years. The minimum probability of 0.02186278 occurs in the year 2013. This value translates to at least one collision occurring in 45 years.

A comparison of the above values indicates that, as might be expected, the highest untracked debris population yields the greatest probability. However, only a total of four years separates all three maximum collision rates. The minimum probabilities of collision values occur in roughly the same time period, again illustrating the general trend of changes within

the middle altitude debris population. The minimum value actually occurs in the model with the highest starting untracked debris population. The interaction of orbital decay with these large numbers of small objects more susceptible to decay may explain the higher rate at which the middle altitude band debris density decreases.

The most important point to make about the Space Station collision probabilities does not concern trends in their growth, but rather the absolute magnitude of their values with regard to Space Station survivability. Clearly, the uncertainty concerning the nature of the untracked debris population is of great importance. My results indicate that even if the magnitude of this population is no more than three times that of the tracked population, at least one collision can occur in as little as 20 years. For a spacecraft that is being designed to operate indefinitely, this figure indicates that a problem will exist for the Space Station barely after it has gotten off the ground. Analysis of the number of close encounters with debris, discussed in the next section, provides an alternate way of looking at the problem and even more startling results.

Debris Encounter Analysis

The analysis of the number of debris encounters

using the method of calculation discussed in chapter four involves the observation of general trends over the years of calculation, and the comparison within and between models of different starting untracked debris populations of encounter values obtained using varying buffer zones. The following paragraphs cover each of these points of analysis in more detail.

To make the following discussion less cumbersome, I will use the following notation to describe the various models incorporating varying buffer zones and starting untrackable debris populations. For example, 3x1km describes the model incorporating a starting untrackable debris population three times the trackable debris population and a one kilometer radius buffer zone surrounding the Space Station. Therefore, 15 models comparing the debris encounters along with the untracked debris populations are identified as follows:

3x1km	5x1km	8x1km
3x3km	5x3km	8x3km
3x5km	5x5km	8x5km
3x7km	5x7km	8x7km
3x10km	5x10km	8x10km

I used the baseline value of a 10 kilometer radius buffer zone to correspond to initial NASA estimates, and incorporated the smaller buffer zones arbitrarily to check the sensitivity of debris encounters to this

parameter.

Since the debris encounter calculation used hinges upon the debris spatial density, the trend observed matches that found for the collision probability calculations. Figures 6.2, 6.3, and 6.4, present several examples representative of these trends by buffer zone and starting untracked debris population. Notice that the number of encounters generally drops off faster over the years for those models incorporating a higher starting untrackable debris population. Since both the collision probability and debris encounter calculations are so closely tied to the debris spatial density, it seems logical to attribute the higher rate of decrease in debris encounters to the interaction between small launch rates and a greater decaying small object population out of the medium band, as was proposed for the trends in the collision probability. While trends illustrate the relationship between the number of encounters and the debris population, analysis of specific magnitudes provides a more exact indication of the severity of the debris environment and its impact on Space Station operations.

The analysis of tracked and untracked debris encounters in a debris environment initially containing untrackable debris three times the trackable debris

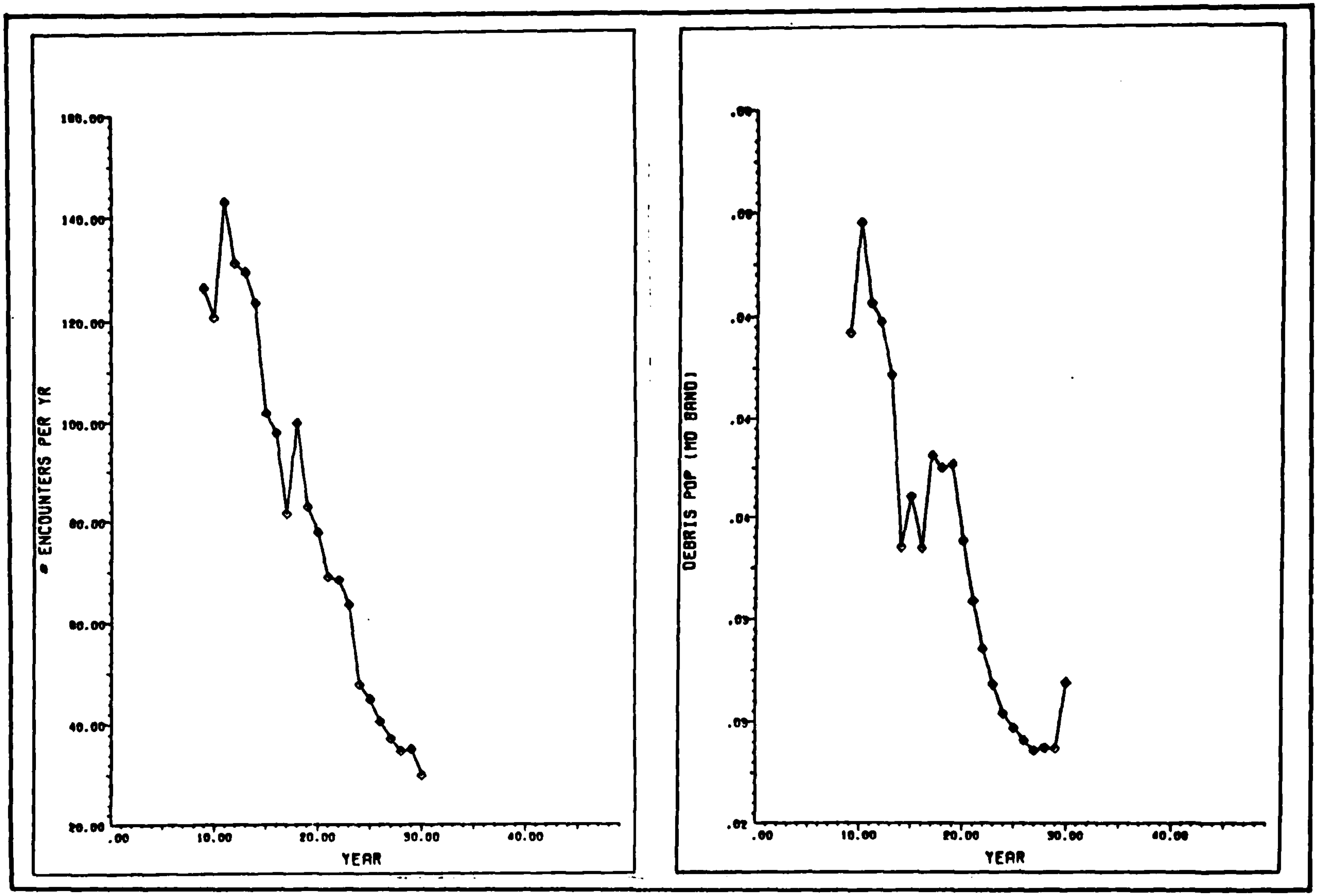


Figure 6.2. Trend of Debris Encounters per Year and Debris Population Over Time for 3.1km Model

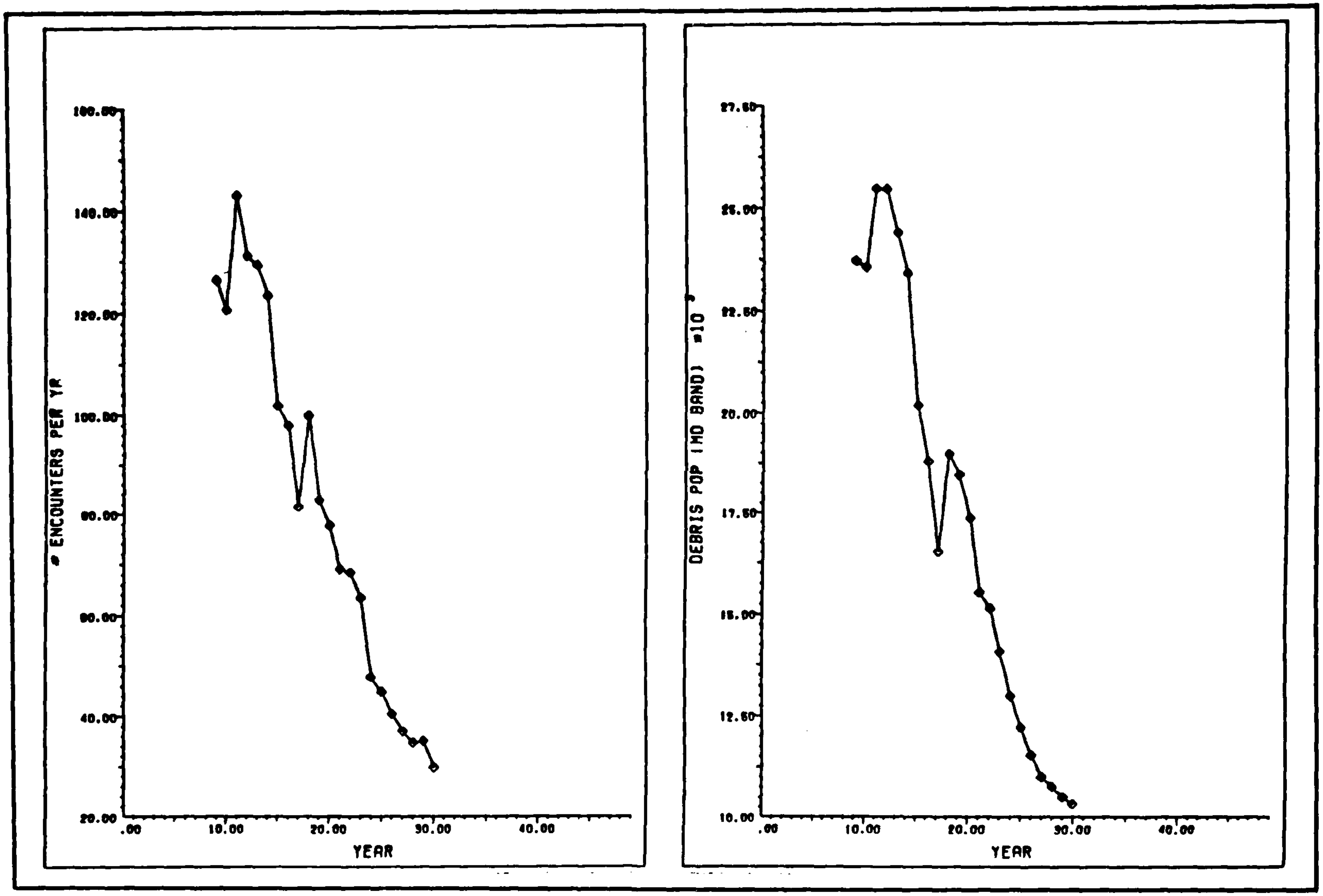


Figure 6.3. Trend of Debris Encounters per Year and Debris Population Over Time for 5.1km Model

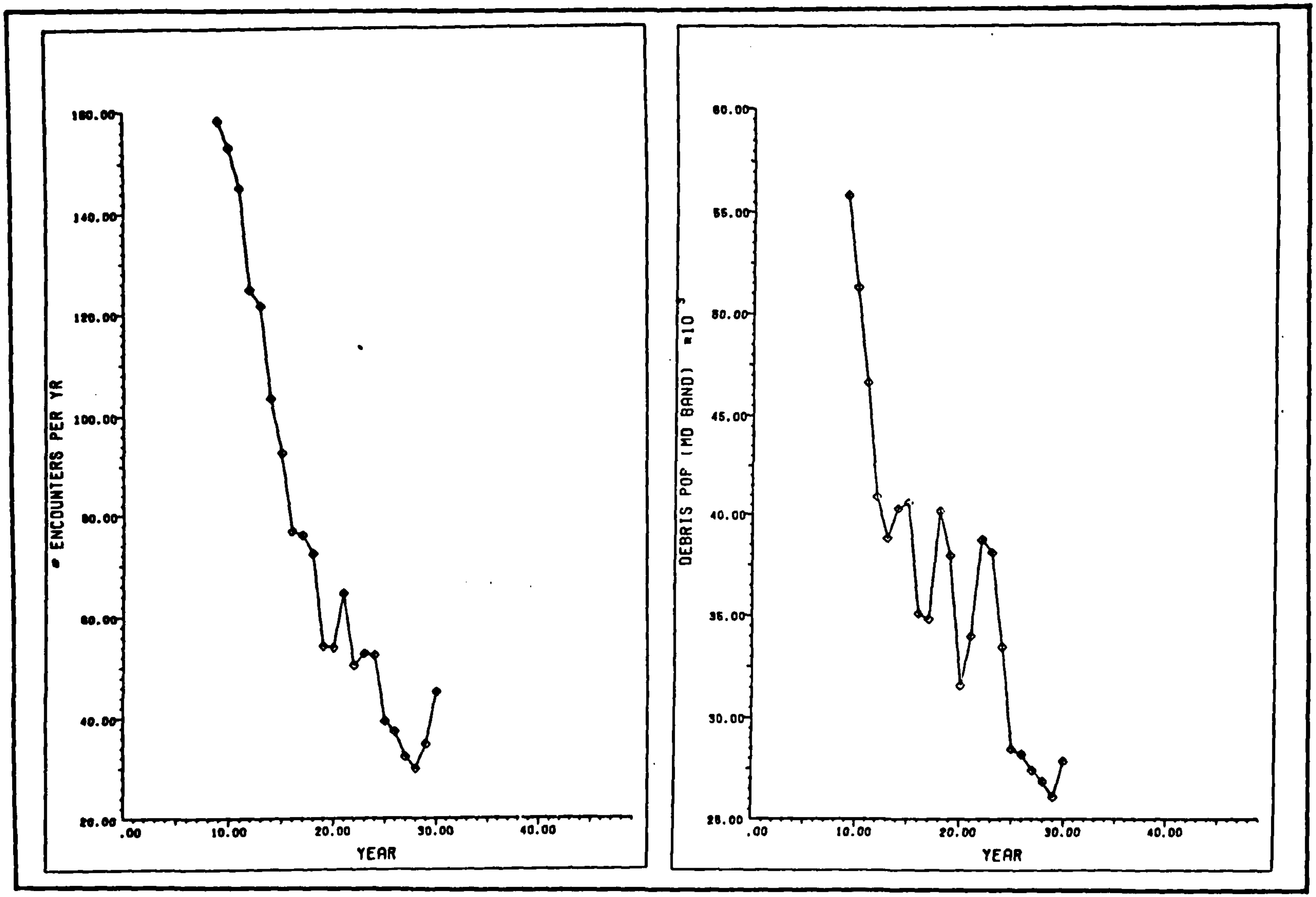


Figure 6.4. Trend of Debris Encounters per Year and Debris Population Over Time for 8.1km Model

population yields startling results, as shown in Table 6.2. The baseline 3x10km model generates an average of around 9,251 encounters with trackable and untrackable debris per year, compared to 4,520 for the 3x7km model, 2,293 for the 3x5km model, 810 for the 3x3km model, and only 67 for the 3x1km model. These quantum leaps in the number of debris encounters can be put another way. The 3x3km encounters are 12 times the 3x1km encounters. The 3x3km encounters, in turn, are close to three times smaller than 3x5km average. Likewise, the 3x7km average is approximately twice as large as the 3x5km number. Finally, the 3x10km average is also two times the 3x7km number. Overall, the 3x10km encounter average is 138 times as large as the 3x1km average.

Comparison of the difference in the encounters experienced in the other models varying the starting untrackable debris populations yield similar results, albeit at much greater orders of magnitude. Tables D.1 and D.2 present these results. Approximately the same orders of magnitude between the average number of encounters per year for each buffer zone hold for the models considering five times and eight times the trackable debris as they did for the models discussed in the previous paragraphs.

The comparison of trackable and untrackable debris

TABLE 6.2

Debris Encounters Per Year for 3x.. Models Varying Buffer Zone Radius

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	105.56	1153.44	3248.89	6391.22	13073.00
1993	117.00	1283.56	3612.78	7106.44	14529.56
1994	104.33	1165.89	3283.67	6458.89	13208.33
1995	97.11	1132.33	3186.44	6271.22	12825.00
1996	94.00	1051.67	2962.67	5831.44	11928.56
1997	84.00	940.78	2664.78	5248.67	10738.00
1998	69.33	868.78	2460.11	4845.00	9914.78
1999	66.78	796.89	2253.22	4442.89	9095.78
2000	79.78	902.89	2553.33	5028.78	10289.78
2001	78.67	885.56	2497.89	4922.44	10074.22
2002	71.22	886.67	2510.22	4944.67	10116.89
2003	69.11	790.00	2241.44	4422.56	9050.89
2004	60.44	719.00	2039.22	4027.22	8242.89
2005	57.78	660.00	1879.89	3707.67	7598.67
2006	51.56	621.33	1763.44	3483.11	7134.33
2007	40.56	583.67	1667.89	3290.44	6749.22
2008	40.44	566.00	1619.11	3201.78	6554.89
2009	40.44	553.44	1578.56	3121.78	6394.67
2010	39.67	543.33	1547.00	3054.78	6260.00
2011	34.67	541.56	1556.78	3070.56	6296.22
2012	34.67	545.78	1549.44	3069.00	6284.56
2013	40.11	617.33	1774.11	3495.56	7163.22
$\bar{x}$	67.15	809.54	2293.22	4519.82	9251.07
σ	25.51	235.59	654.74	1283.14	2619.03

encounters between models of varying untracked debris starting populations using the same size buffer zone are illustrated in Tables 6.3 and D.3 through D.6. In all cases, the models with the starting untrackable debris population five times the trackable population yield the highest average number of encounters per year averaged over all years of interest. Figures 6.5 through 6.9 illustrate this conclusion in graphical form.

As is believed to be true for the collision probabilities, the apparent discrepancy between the 5x.. and 8x.. model encounters is probably caused by increased flow out of the medium band of smaller particles characterizing the higher untracked debris populations. Overall, the number of encounters between the models for each buffer zone are quite similar. On the average, the number of encounters per year for the 3x.. models are 84.4% of the 5x.. models. The 8x.. models maintain an average yearly number of encounters 93.6% of the 5x.. models. While a discussion of yearly magnitudes of debris encounters is helpful in introducing the possible severity of the problem, a specific look at the number of encounters occurring between scheduled refueling of the Space Station brings home the impact of the debris environment on this spacecraft's operations.

TABLE 6.3
Debris Encounters Per Year for Models Using 1 km Buffer Zone and Varying Untracked Debris Populations

YEAR	UNTRACKED DEBRIS POPULATION		
	3X	5X	8X
1992	105.56	126.33	158.22
1993	117.00	120.56	153.11
1994	104.33	143.00	145.11
1995	79.11	131.00	124.89
1996	94.00	129.22	121.56
1997	84.00	123.33	103.44
1998	69.33	101.78	92.67
1999	66.78	97.78	76.78
2000	79.78	81.33	75.89
2001	78.67	99.78	72.22
2002	71.22	82.67	54.00
2003	69.11	77.56	53.78
2004	60.44	68.89	64.44
2005	57.78	68.22	50.22
2006	51.56	63.33	52.56
2007	40.56	47.67	52.22
2008	40.44	44.78	39.33
2009	40.44	40.44	37.11
2010	39.67	37.00	32.11
2011	34.67	34.67	29.78
2012	34.67	35.00	34.67
2013	40.11	29.78	45.00
$\bar{x}$	67.15	81.10	75.87
σ	25.51	36.95	41.10

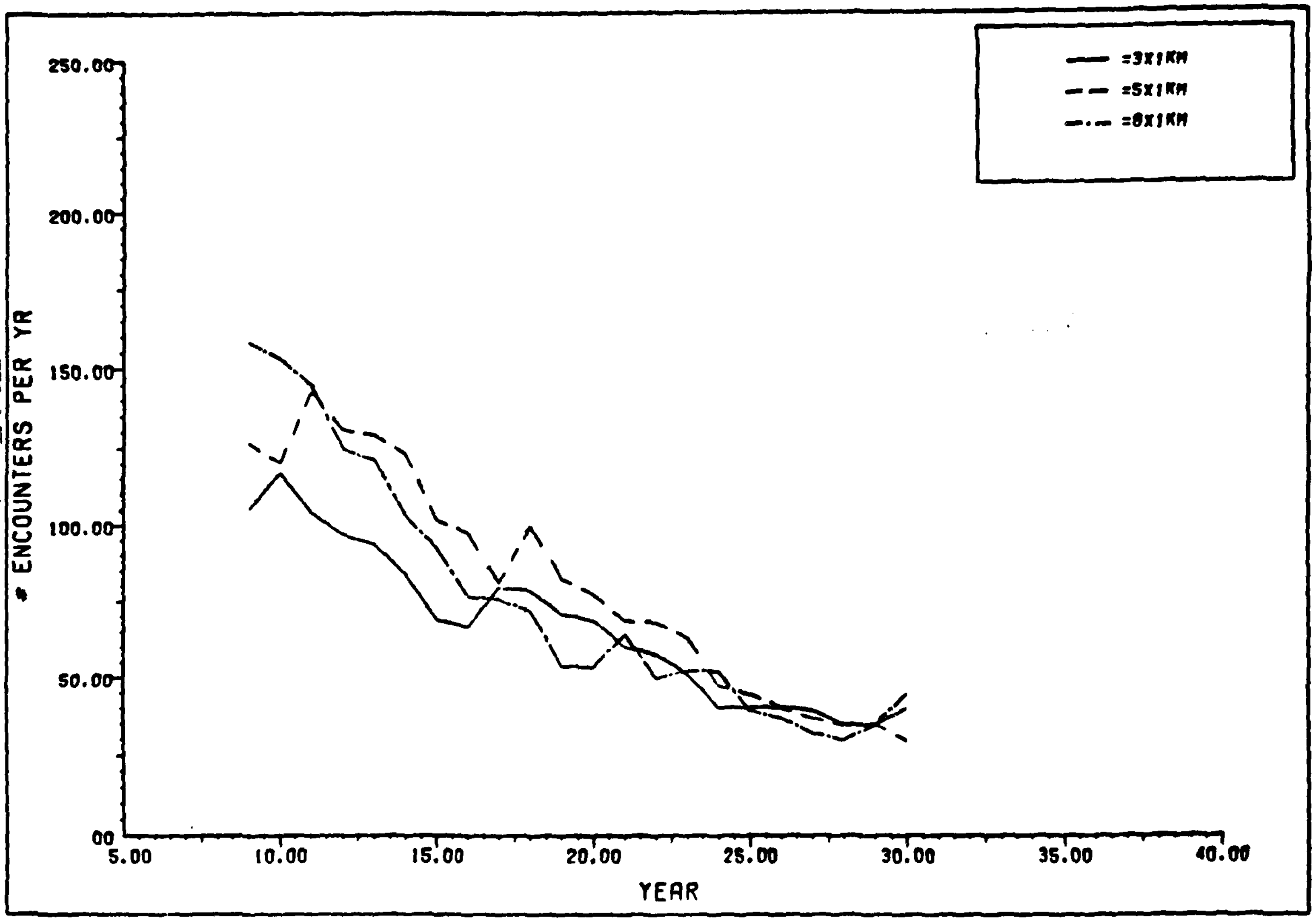


Figure 6.5. Comparison of Debris Encounters per Year for Models Using 1km Buffer Zone and Varying Untracked Debris Populations

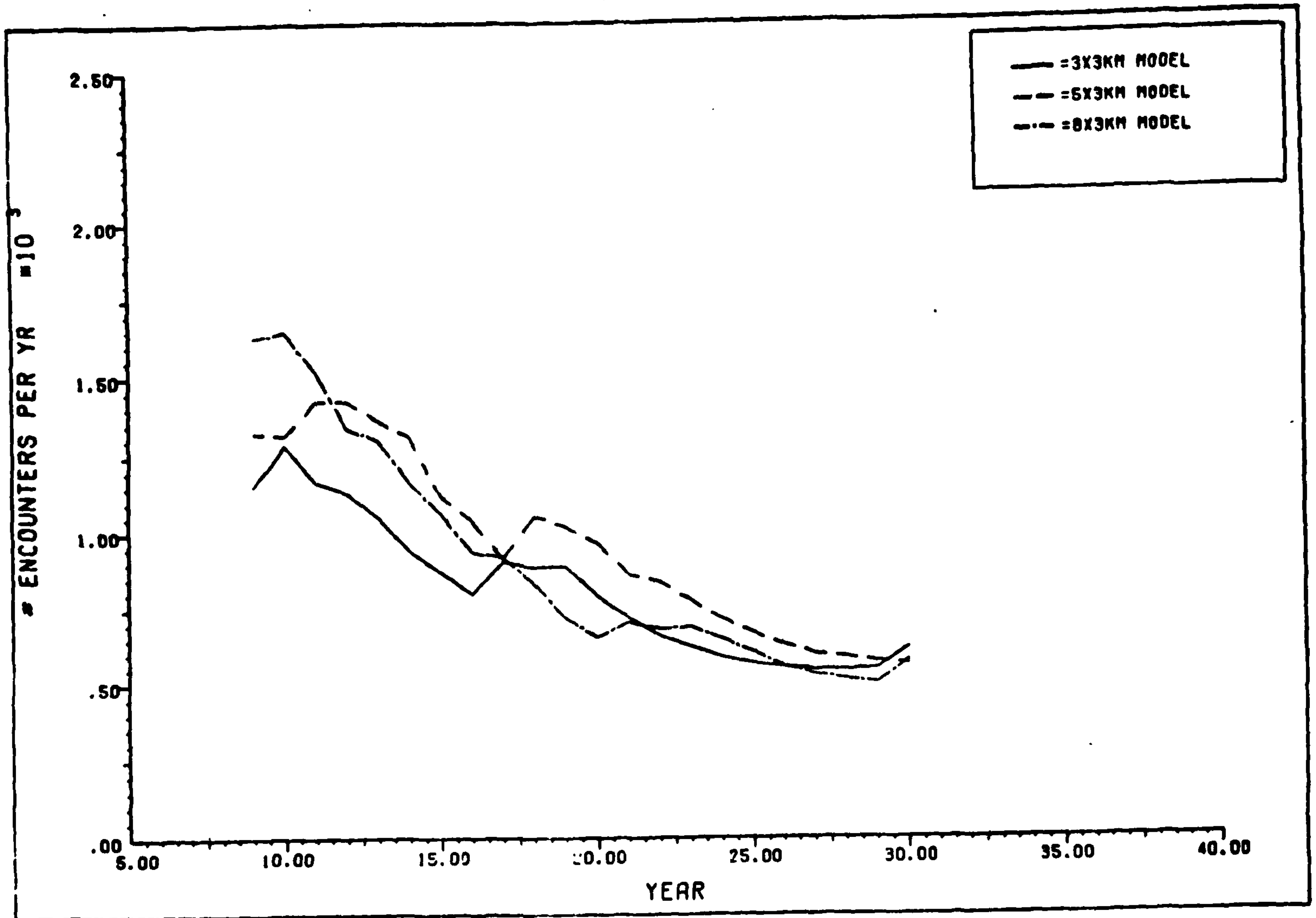


Figure 6.6. Comparison of Debris Encounters per Year for Models Using 3km Buffer Zone and Varying Untracked Debris Populations

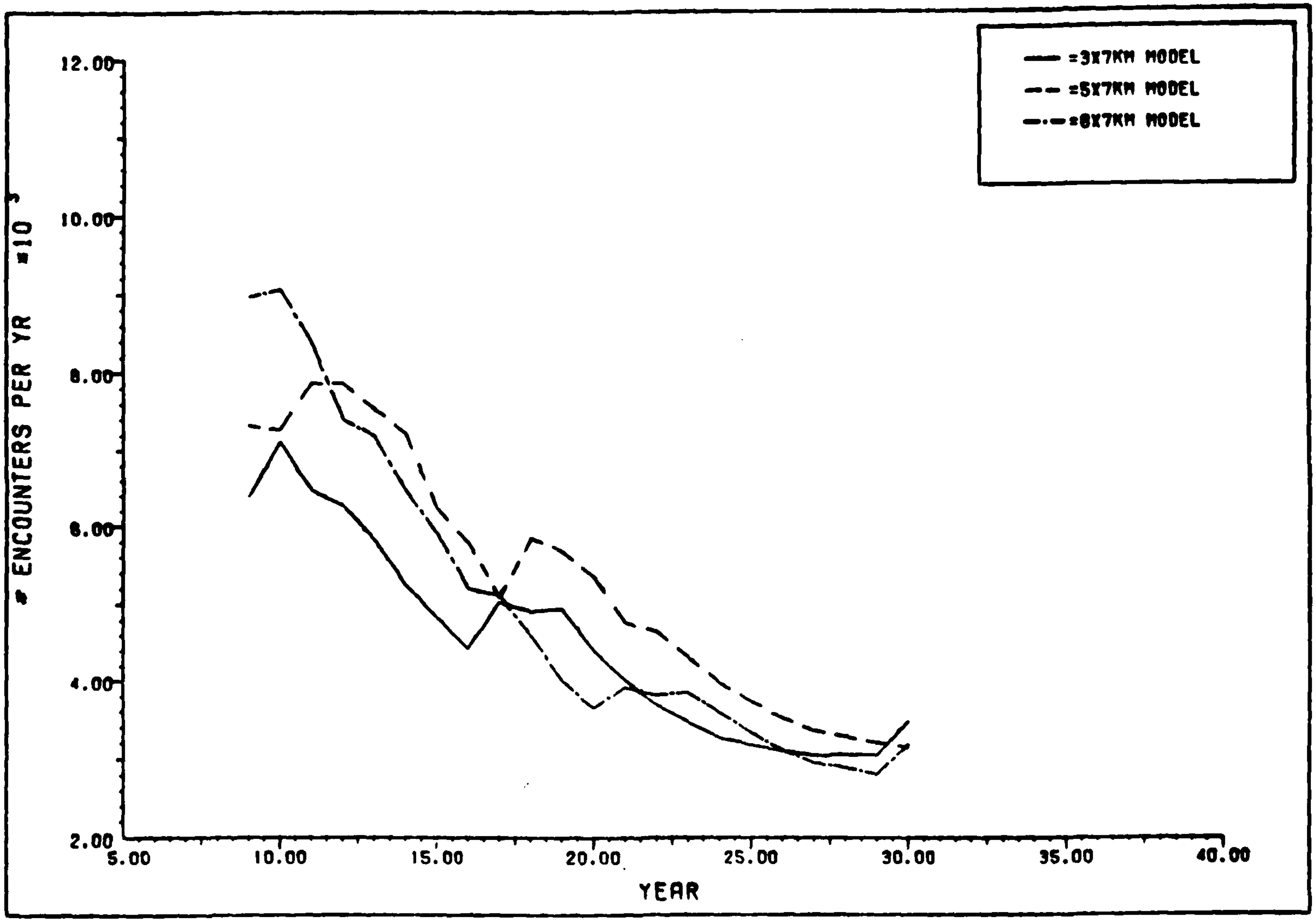


Figure 6.7. Comparison of Debris Encounters per Year for Models Using 5km Buffer Zone and Varying Untracked Debris Populations

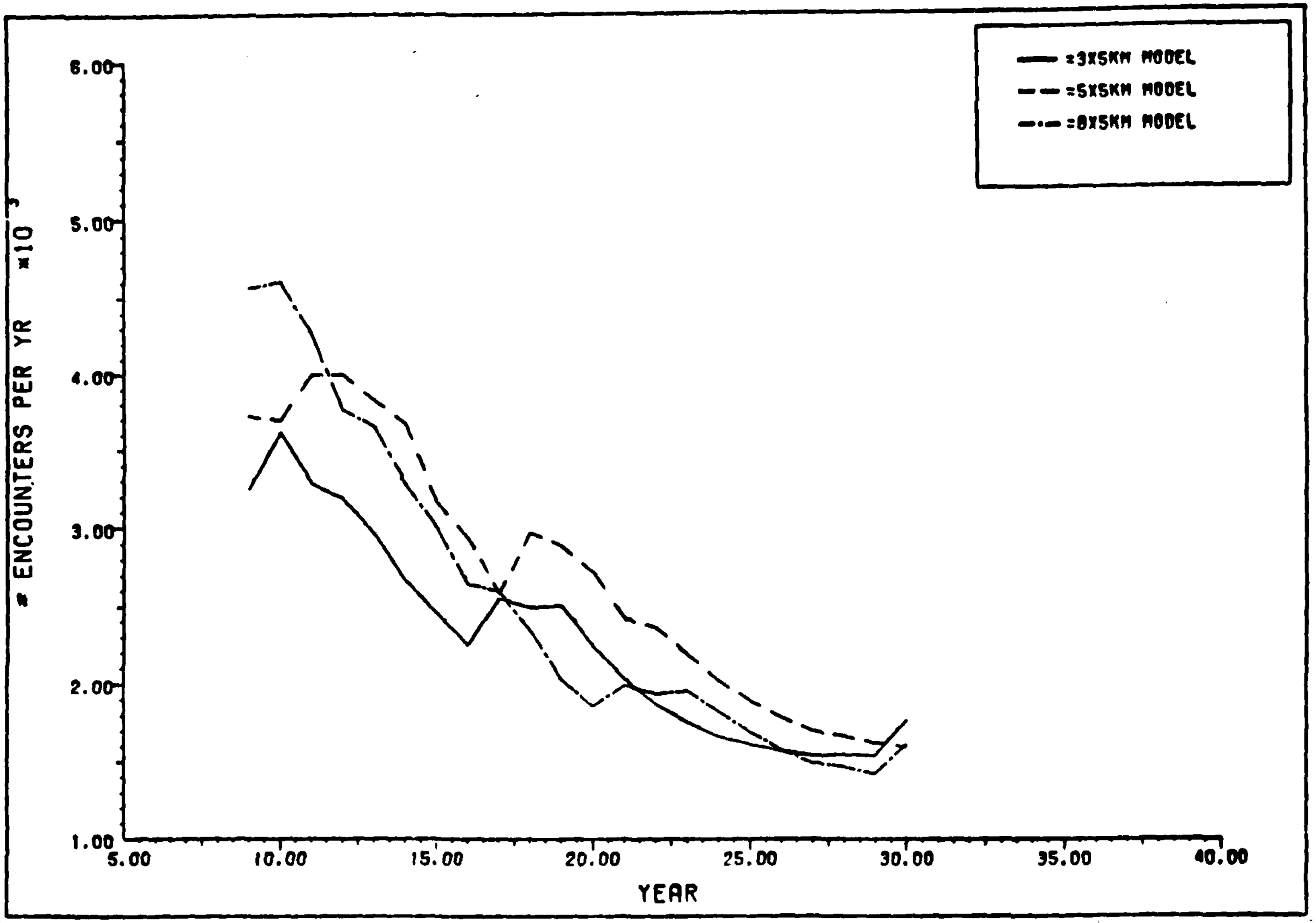


Figure 6.8. Comparison of Debris Encounters per Year for Models Using 7km Buffer Zone and Varying Untracked Debris Populations

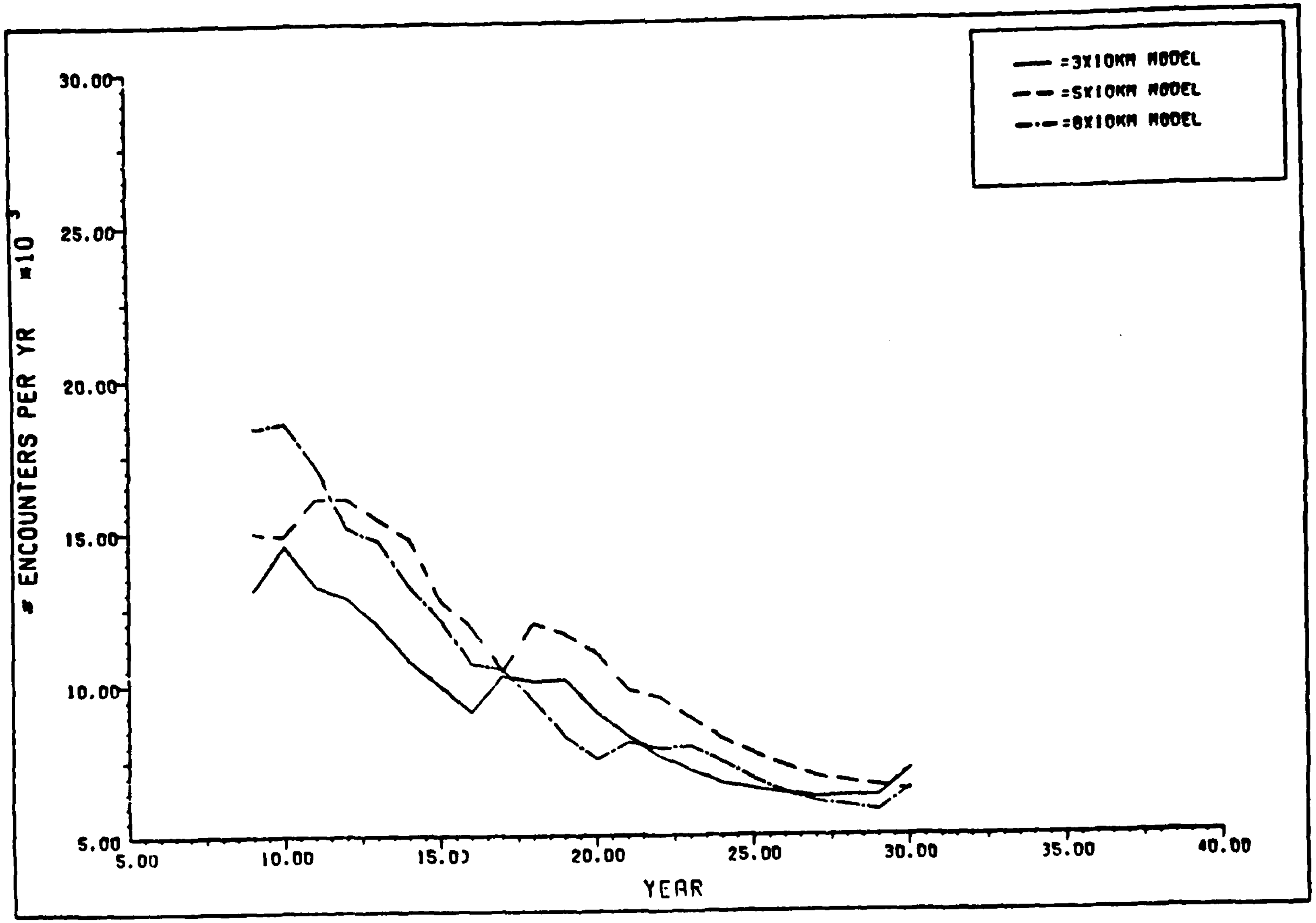


Figure 5.9. Comparison of Debris Encounters per Year for Models Using 10km Buffer Zone and Varying Untracked Debris Populations

Fuel Requirement Analysis. The impact of the debris environment on Space Station fuel requirements comes into play when determining the number of objects invading the established buffer zone over the 90 days between scheduled refuelings. That number of encounters translates to the requirement that the Space Station perform the same number of some type of avoidance maneuver. The maneuver can either be a change in altitude or a change in velocity, thereby erradicating the possibility that the particular debris object and the Space Station orbits intersect at the same time. I only considered change in velocity manuevers in the subsequent calculations since these manuevers require less energy (33).

The magnitude of the maneuver is a function of the desired miss distance and the time the maneuver is initiated before the predicted collision. I arbitrarily chose a 10 kilometer miss distance and a one day advance notification of a close encounter. Shorter miss distances and longer periods of time to perform the maneuver would decrease the required magnitude of the maneuver. Using the above parameters, the required change in velocity calculation is as follows:

10km/1 day = (10,000m/86,400 sec) = 0.1157407 m/sec

This means that if the Space Station changes its

circular orbital velocity by the above amount one day in advance of the predicted encounter, it will be 10 kilometers away from the object at the predicted encounter time.

The number of maneuvers available to the Space Station over a 90-day period is, of course, constrained by its fuel capacity. Therefore, fuel requirements will be in terms of the change of velocity required due to encounters versus the total change in velocity allowed. The maximum change in velocity allowed is a function of the Space Station mass and the total propulsion impulse required over the time period of interest. The initial and final estimated masses of the Space Station over its growth period are 257,870 and 681,320 pounds, respectively (19:63). Conversion of these values to equivalent values in slugs yields 8,008.3581 and 21,159.006 slugs, respectively. The actual calculation cf the maximum allowable change in velocity entails dividing the above values into their respective total propulsion impulse requirements over a 90-day period. For example, the calculation for the Space Station when it is initially deployed is:

$$\frac{160{,}273.97 \text{ lb-sec/90-day}}{8{,}008.3851 \text{ slugs}} = 20.01327 \text{ m/sec/90-day}$$

Likewise, the maximum allowable change in velocity at system maturity is 6.3860886 m/sec/90-day.

Using the required change in velocity calculated earlier, I determine the maximum allowable number of maneuvers at initial system deployment to be:

$$\frac{20.01327 \text{ m/sec/90-day}}{0.1157407 \text{ m/sec/encounter}} = 172.91471 \text{ encounters/90-day}$$

The same type of calculation yields 55.175825 maximum allowable encounters per 90-day basis at system maturity.

Tables 6.4, D.7, and D.8 present the quarterly maximum debris encounters for each year for the fifteen models runs with varying buffer zones and initial untrackable debris populations. The one kilometer radius buffer zone is the only zone in which avoidance maneuvers do not impact the amount of fuel required. The results using the larger buffer zones indicate that the fuel required to perform the maneuvers must be many times the current supply. For example, the maximum number of debris encounters for the 3x10km model occurs in 1995 with a value of 6,325. Accounting for the increased mass of the Space Station at that point in its construction, which I assume increases at a constant rate over the spacecraft's growth period, the number of encounters is still 68 times the maximum allowable number of 92.98 maneuvers.

The apparent unreasonableness of the debris encounters as calculated warrants an investigation into why this is so. Recall that the debris encounter

TABLE 6.4

Maxium Encounters Per Quarter for 3x.. Models Varying Buffer Zone Radius

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	42	449	1257	2471	5048
1993	52	526	1471	2888	5902
1994	52	490	1373	2700	5515
1995	61	564	1576	3099	6325
1996	52	508	1423	2795	5710
1997	39	416	1166	2290	4681
1998	34	346	975	1917	3919
1999	40	451	1266	2487	5082
2000	39	434	1213	2384	4873
2001	52	526	1472	2888	5905
2002	52	475	1334	2622	5359
2003	39	399	1119	2201	4498
2004	26	334	941	1851	3784
2005	26	283	796	1566	3202
2006	26	245	685	1352	2764
2007	13	210	603	1189	2434
2008	13	208	580	1143	2338
2009	13	195	559	1092	2241
2010	13	194	535	1061	2169
2011	13	182	528	1040	2128
2012	13	195	546	1082	2221
2013	26	324	912	1793	3666
$\bar{x}$	33.45	361.55	1015	1995.95	4080.18
σ	16.06	130.14	361.96	709.37	1447.16

calculation involved the sweeping out of a volume by the Space Station with its buffer zone, and counting the number of debris objects lying within that volume over the time period of interest. The possible problems with that calculation when operating within the parametric model are that (1) no consideration is given to relative velocities between the Space Station and the debris, (2) objects can be repeatedly encountered, and (3) the model incorporates the assumption that all objects are uniformly distributed within each altitude band. The interaction of these factors would cause a much higher count to be obtained. Therefore, I set out to look for an alternate method of calculating the number of debris encounters.

<u>Alternate Encounter Calculation Development</u>. I believed that the lack of consideration given to relative velocities in the debris encounters calculation was the single factor that could be most easily rectified and provide more realistic encounter values. Therefore, I turned my attention back to the Poisson distribution where the distribution parameter developed in chapter four did in fact include relative velocity. I reasoned that I could repeatedly use this distribution to calculate the probability of n or less encounters until a certain desired probability was

reached. This would essentially give me the worst case value at a desired confidence level, which I chose to be 97.5%.

The actual calculations involved using the S statistical package to generate Poisson probabilities by way of the gamma distribution until the minimum cumulative probability above 0.975 was reached. The value of n used in the calculation was then selected as representing the highest number of encounters to be expected with that particular debris population. I used the average yearly debris populations obtained from the model output as inputs into this alternate calculation.

Tables 6.5, D.9, and D.10 present the results of these calculations. The alternate calculation technique does decrease the number of encounters for the larger buffer zones. However, it does not change the result for the one kilometer radius buffer zone where encounters fall below the maximum allowable number of encounters. The alternate method actually calculates the same or greater number of encounters for the model incorporating a one kilometer buffer zone, but as the size of the buffer zone increases, the number of encounters drops off from the original calculation dramatically. For example, the encounter calculation using the Poisson distribution generates

TABLE 6.5
Debris Encounter Calculations Using Poisson Distribution for 3x.. Models
Varying Buffer Zone Radius

YEAR	BUFFER ZONE RADIUS (KM) 1 km	3 km	5 km	7 km	10 km
1992	52	273	650	1183	2288
1993	65	299	715	1313	2522
1994	52	273	663	1196	2301
1995	52	273	637	1170	2249
1996	52	260	598	1092	2093
1997	52	234	546	1001	1911
1998	52	221	520	923	1768
1999	39	208	481	858	1638
2000	52	234	533	962	1833
2011	52	221	520	936	1794
2002	52	221	520	949	1807
2003	39	208	481	858	1625
2004	39	195	442	793	1495
2005	39	182	416	728	1391
2006	39	169	390	689	1313
2007	39	169	377	663	1248
2008	39	156	364	650	1209
2009	39	156	351	637	1183
2010	39	156	351	624	1170
2011	39	156	351	624	1170
2012	39	156	351	624	1170
2013	39	169	390	702	1313
$\bar{x}$	45.5	208.59	483.95	871.59	1658.68
σ	7.77	46	115.51	217.48	430.39

only 35.3% as many maximum encounters in a 90-day period on the average as the original calculation does for the 8x10km model. Figures 6.10 through 6.12 provide examples showing this trend for the 3x1km, 3x5km, and 3x10km models.

Sensitivity Analysis

In addition to the analysis discussed thus far involving models of different starting untracked debris populations, I performed additional sensitivity analysis on what I felt were two other important system elements affecting the probability of collision. The elements of launches and potentially explodable objects were chosen because the first is the major factor controlling the debris distribution and the second is the primary contributor to the debris populations. As the causal diagram discussed in chapter three indicates, these system elements are linked together since launches are the only sources of potentially explodable objects.

Besides selecting these system elements for sensitivity analysis based on their importance in the space debris environment system, I also chose them for the very real possibility that the parameters associated with these elements would change over the years. The parametric model used up to this point in the analysis kept both the launch rate and the number

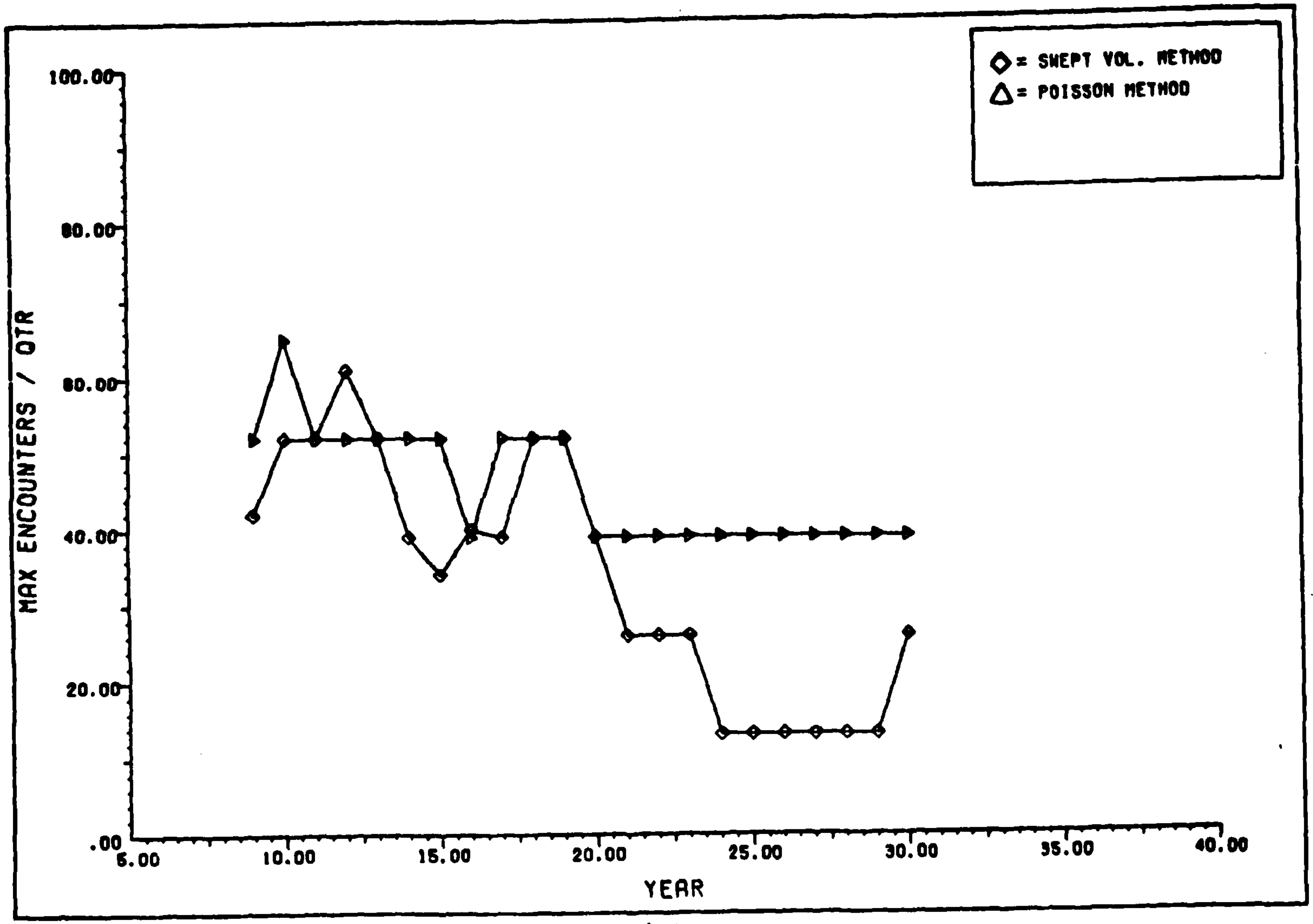


Figure 6.10. Comparison of Maximum Encounters per Quarter for 3x.. Models using 1km Buffer Zone and Different Encounter Calculation Methods

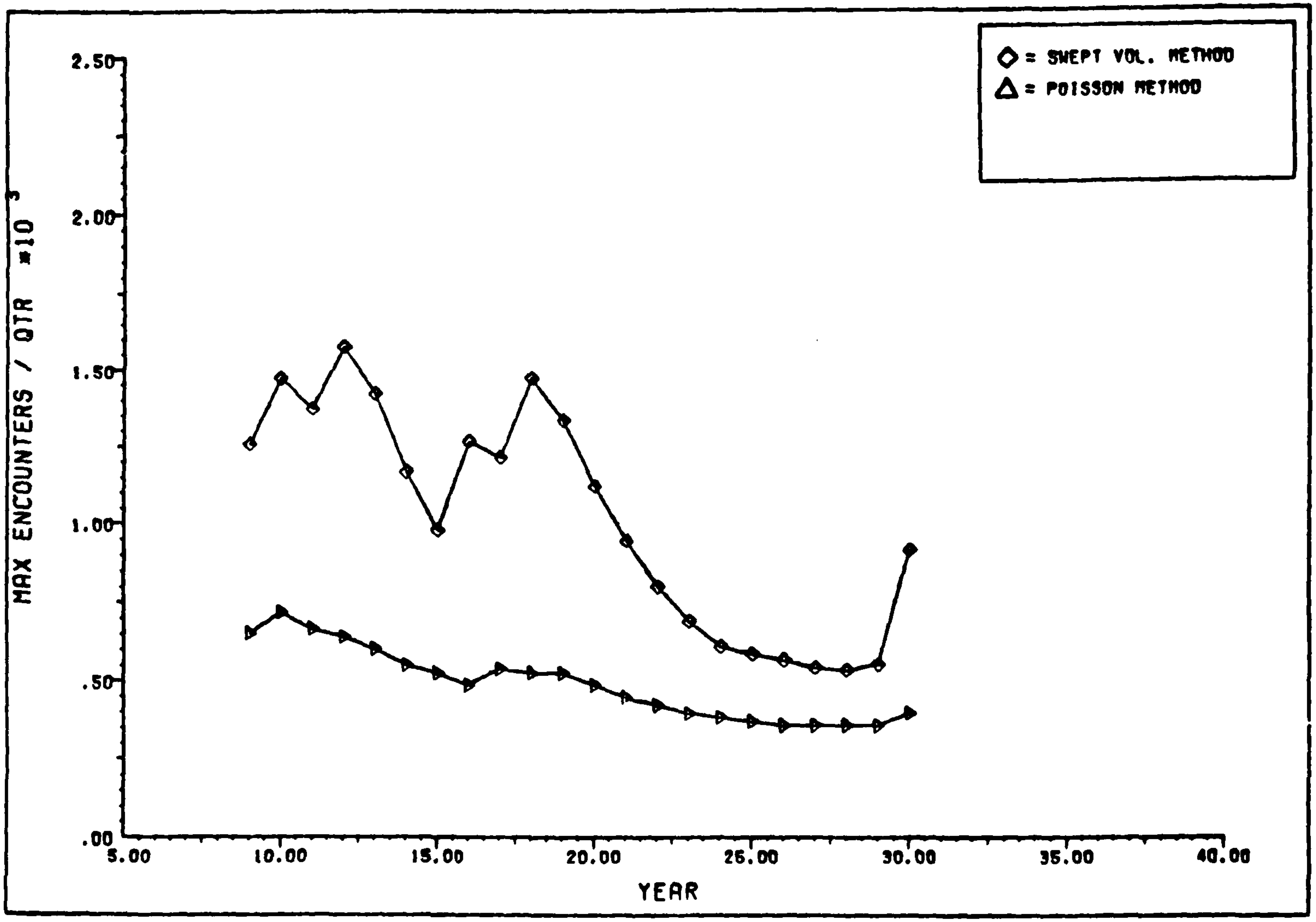


Figure 6.11. Comparison of Maximum Encounters per Quarter for 3x.. Models using 5km Buffer Zone and Different Encounter Calculation Methods

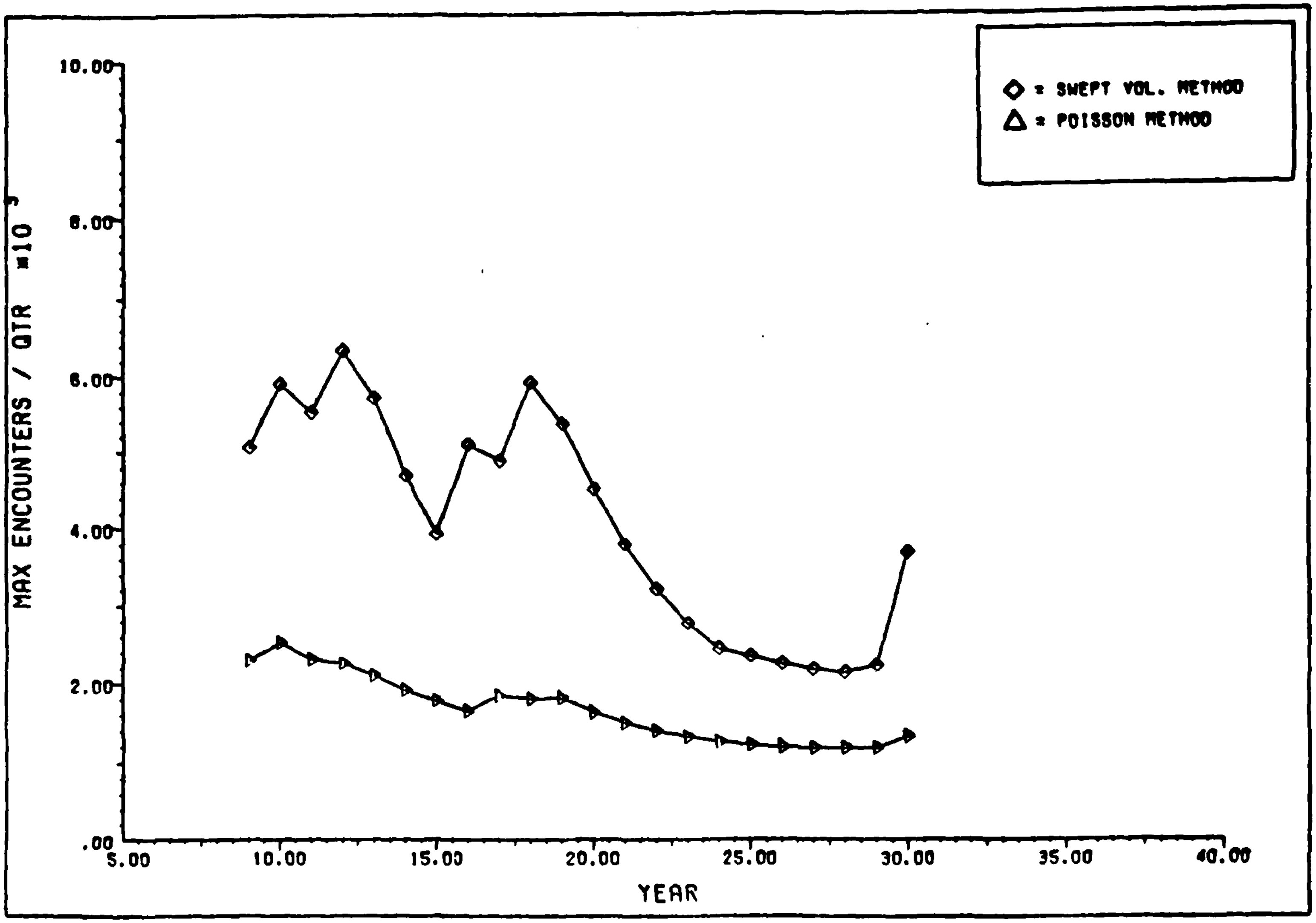


Figure 6.12. Comparison of Maximum Encounters per Quarter for 3x.. Models using 10km Buffer Zone and Different Encounter Calculation Methods

of potentially explodable objects deposited by a launch constant throughout the simulation. However, the growing interest of many nations in space may in fact cause the launch rate to increase in the future. Also, experience and better engineering over the years may decrease or completely eradicate the placement of potentially explodable objects in space.

Concrete data did not exist on which I could base my changing the parameters of these system elements within the model. Historical launch rates have indeed grown over the years, but that growth has been erratic and primarily tied to the two superpowers. Therefore, I chose to adjust the average launch rate upward by two each year as well as the maximum and minimum number of launches allowed for that year. I used the variables LMEAN, LMAX, and LMIN for this purpose.

There was also no way of knowing exactly when potentially explodable objects would begin to decrease due to redesign efforts. In addition, these objects could still be deposited despite our best efforts and by nations new to the space program which do not have the experience, engineering expertise, money, nor time required to work the problem. I therefore chose to treat the number of explodable objects added as a variable using a uniform distribution to generate zero to three explodable objects for each launch. I created

a new variable EXPAD for this purpose.

I used the above changes to the model in various combinations with the varying number of starting untracked debris populations. Either (1) the launch rate was held constant and the explodable objects added was allowed to vary, (2) the launch rate increased and the explodable objects added remained constant, or (3) both were allowed to vary. Discussion of sensitivity analysis results incorporates certain notation describing each of these models, of which examples are 3xliec and 8xlcev. The "3x" and the "8x" indicate the magnitude of the starting untracked debris populations above the tracked populations. The "l" and "e" represent launches and explodable objects deposited, respectively. The "c" indicates that the parameters associated with that particular system element remain unchanged from the baseline model. Finally, an "i" indicates an increasing launch rate and a "v" represents a varying number of explodable objects being deposited.

The results of the sensitivity analysis models are quite significant in showing the impact of launches and potentially explodable objects on the probability of collision. Table 6.6 presents the collision probabilities by year for the baseline, 3xliec, 3xlcev, and 3xliev models. The 3xliec average collision

TABLE 6.6

Collision Probabilities for Models Varying Launch Rate and Number of Explodable Objects Added

YEAR	MODEL TYPES			
	BASELINE	3x1iec	3x1cev	3x1iev
1992	.04417984	.04710845	.03765426	.01600658
1993	.04948522	.05081905	.03271499	.01483373
1994	.04559287	.05215969	.02867474	.01402144
1995	.04469331	.05619323	.05466592	.01360019
1996	.04213160	.05274885	.04701716	.01296066
1997	.03347091	.06033363	.04086820	.01244894
1998	.03599253	.05535555	.03564259	.01221952
1999	.03342433	.05533770	.03164308	.01228196
2000	.03807677	.05586892	.02870997	.01221768
2001	.03743685	.05395369	.02572057	.01222774
2002	.03762065	.05675412	.02349717	.01215591
2003	.03377774	.05518003	.02187713	.01236752
2004	.03084769	.05425863	.02036143	.01231636
2005	.02849567	.05202202	.01898672	.01219805
2006	.02679938	.05458822	.01788025	.01220383
2007	.02537565	.06073925	.01761306	.01217938
2008	.02467216	.06569868	.01733582	.01247737
2009	.02407048	.07237769	.01690777	.01226209
2010	.02358310	.06594069	.01659968	.01206763
2011	.02370759	.06417050	.01607260	.01258639
2012	.02366490	.06316163	.01603832	.01287449
2013	.02687203	.06733568	.01601283	.01286330
$\bar{x}$	.0333624	.0578229	.025304	.0130518
σ	.008222	.0063027	.0121622	.0015964

probability of 0.0578229 translates to a collision rate of at least one in only 17 years. This is 12 years sooner than that predicted by the baseline model. Keeping the launch rate constant while varying the number of potentially explodable objects deposited results in a collision rate over two times greater than that for the baseline model. This is approximately another two times greater than that for the 3xliev model, which predicts an average of at least one collision in approximately 76 years.

The results underscore the significance of the explodable object population and bring to light the relative unimportance of launches in contributing to the debris population. Figure 6.13 compares the explodable object medium altitude band populations for the 3xliec and 3xliev models, and Figure 6.14 presents the corresponding changes in the debris populations. It is apparent that controlling the potentially explodable objects deposited in space stabilizes the debris population even with increasing launch rates. Figure 6.15 illustrates that despite an increasing launch rate, the 3xliev model's medium band debris population is generally less than that of the 3xlcev model.

Tables D.11 and D.12 present the results for the sensitivity analysis models using starting untracked

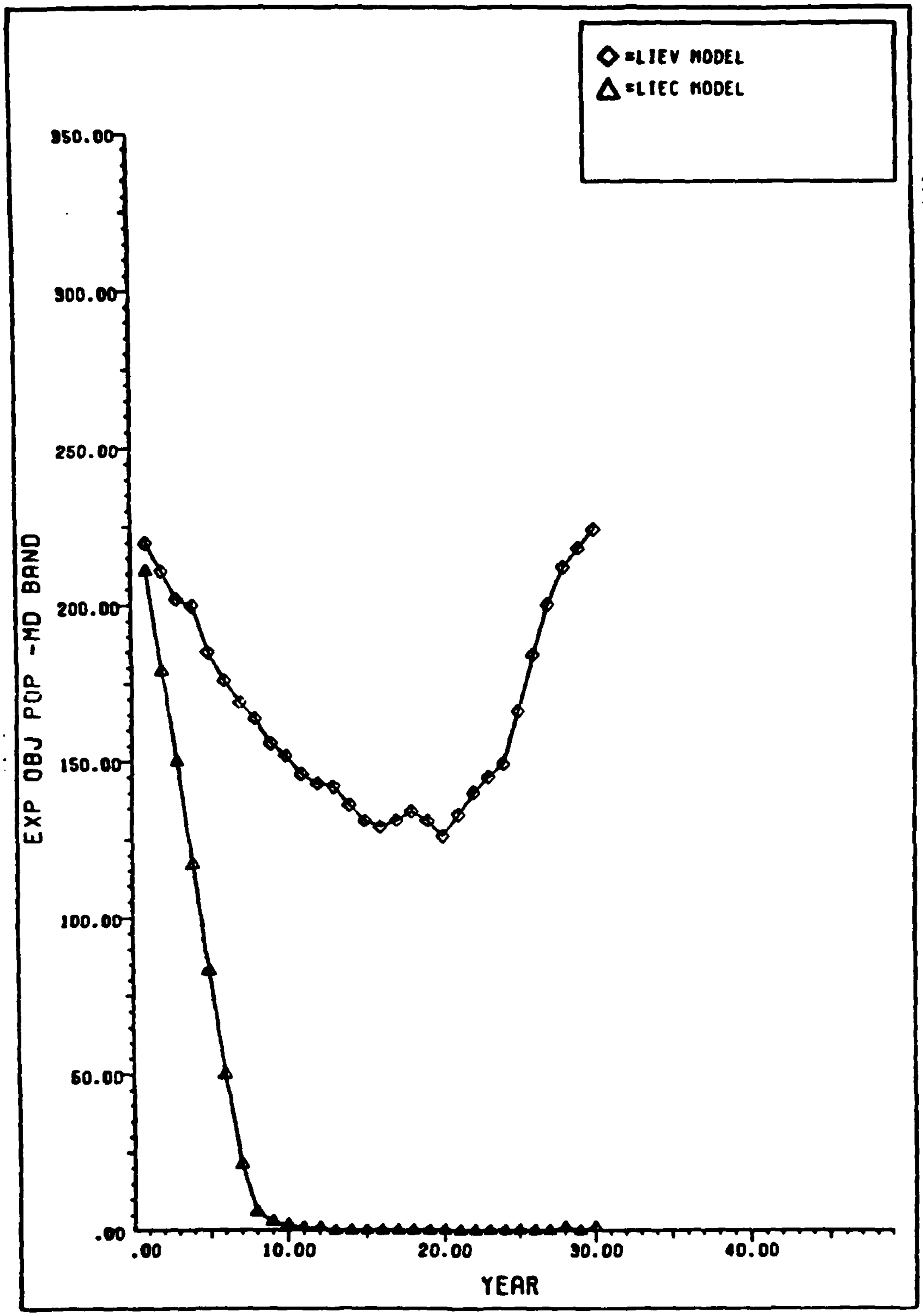


Figure 6.13. Comparison of Medium Altitude Band Explodable Object Populations for 3x.. Models Varying the Number of Explodable Objects Added

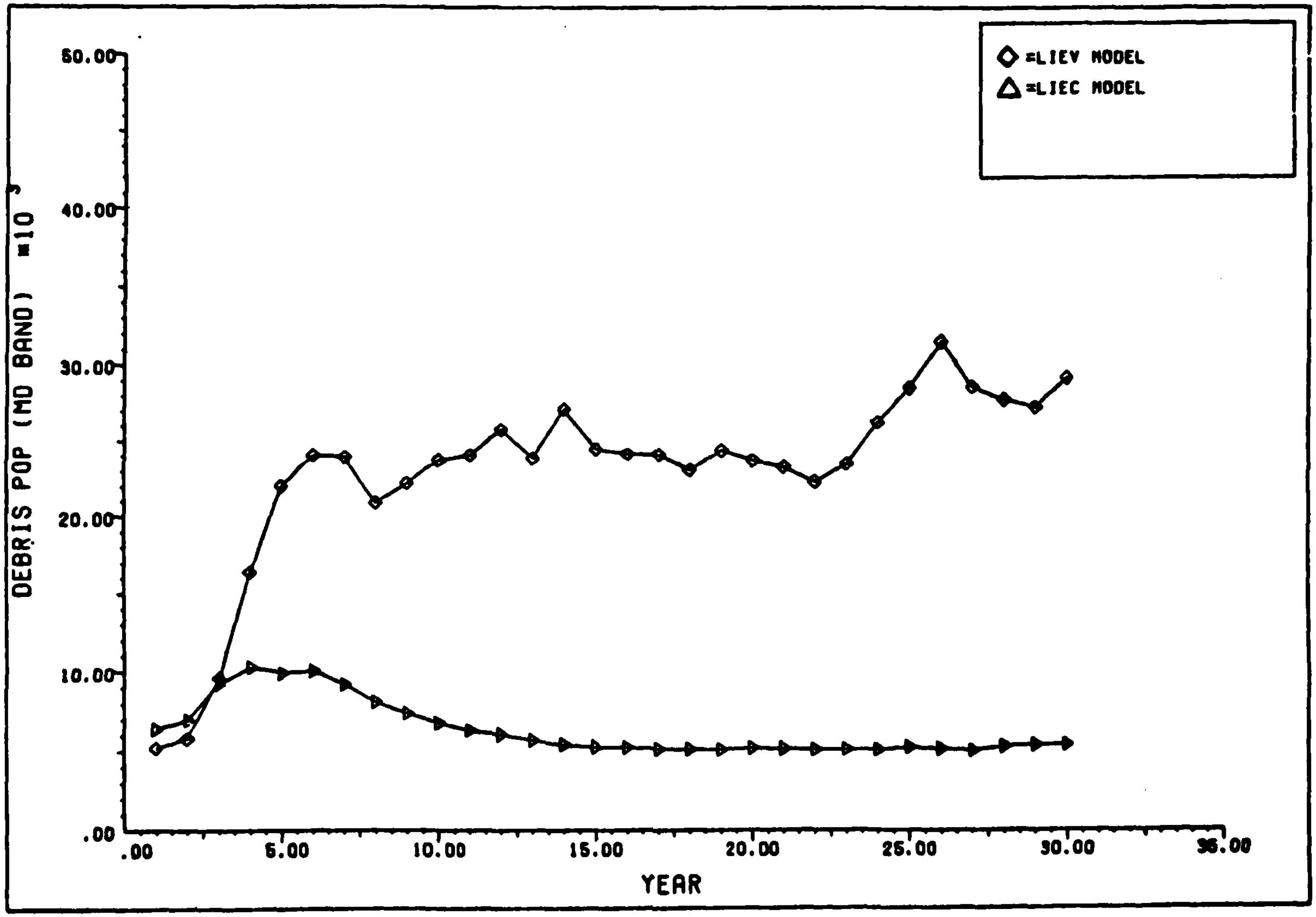


Figure 6.14. Comparison of Medium Altitude Band Debris Populations for 3x.. Models Varying the Number of Explodable Objects Added

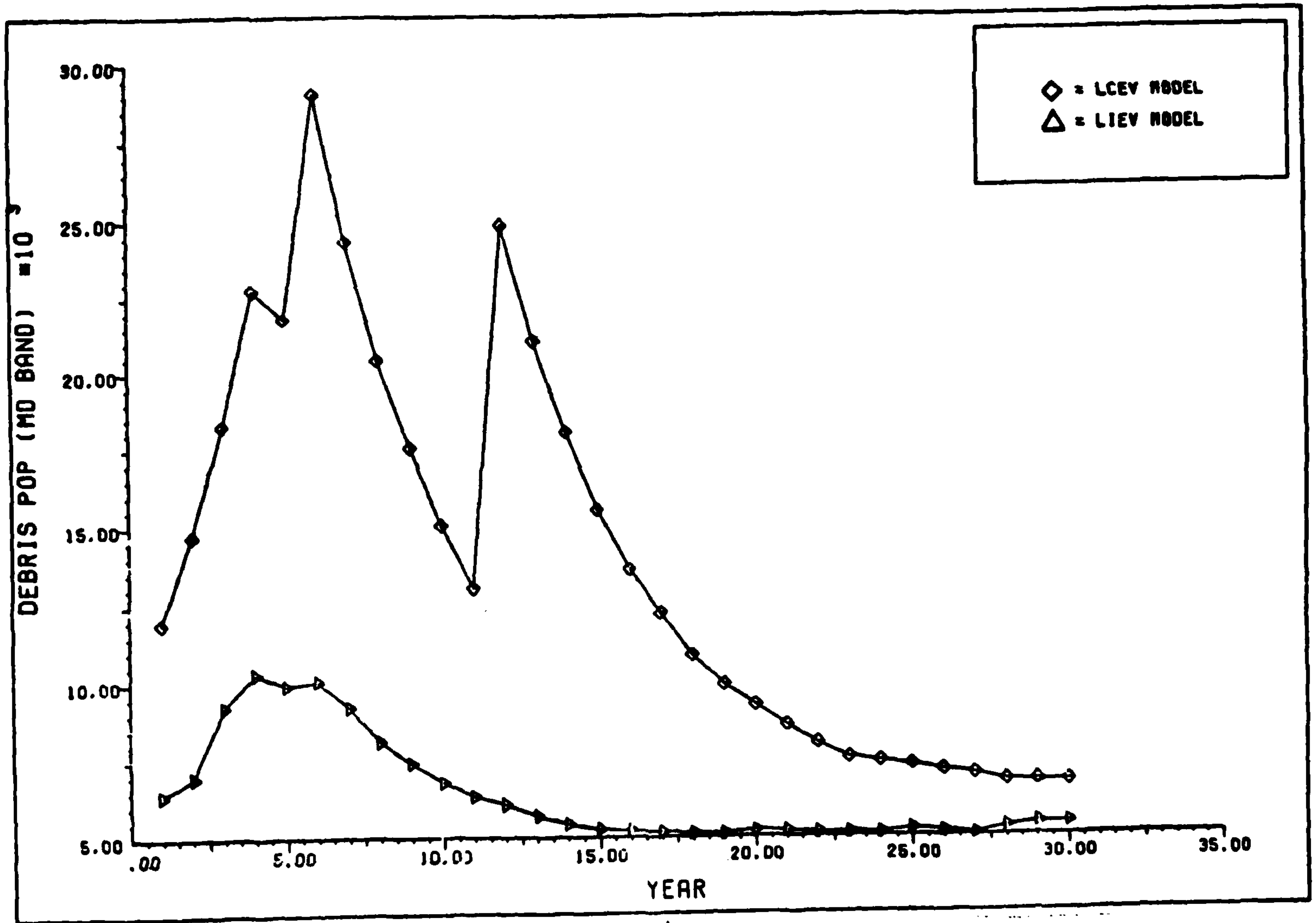


Figure 6.15. Comparison of Medium Altitude Band Debris Populations for 3x.. Models Varying the Launch Rate

debris populations five and eight times the number of trackable objects. The same general relationships between the models exist as they did for the 3x.. models, with the average collision probabilities for the 5xliec and 8xliec models being 50.8 and 63.2% greater than their respective baseline models. As for the 3x.. models, the 5xlcev, 8xlcev, 5xliev, and 8xliev models demonstrate the tremendous effect unintentional explosions have on the debris population, and hence on the Space Station collision probability.

Conclusion

This chapter concentrated on the analysis of the Space Station probability of collision, the number of trackable and untrackable debris encounters, and the affect of certain system elements on the magnitude of the collision probability. Analysis of the collision probability involved the comparison of these values resulting from models incorporating varying untrackable debris populations. Results indicated that the Space Station could expect to collide with an object well within its operational lifetime even if the untrackable debris population was only three times the trackable population.

Analysis of the number of debris encounters involved the comparison of these values resulting from models incorporating varying buffer zones which, when

violated by the entry of a debris object, required that the Space Station perform an avoidance maneuver. The number of encounters calculated using the swept volume technique were quite large, prompting the use of an alternate technique involving the Poisson distribution. However, both methods resulted in the same general conclusion that a buffer zone any larger than a one kilometer radius would require that the Space Station perform avoidance maneuvers beyond that which is allowed by its fuel capacity.

Sensitivity analysis involved two system elements directly under human control--the launch rate and the number of potentially explodable objects deposited per launch. Results showed that the second element was by far the most critical in altering the collision probability. An increase in the number of launches per year combined with no efforts taken to redesign potentially explodable components could result in a debris population large enough to cause a collision with the Space Station in as little as 16 years.

VII. Conclusions and Recommendations

Introduction

The purpose behind this research effort was to give an initial indication of the potential problem involving Space Station operations in the space debris environment. Reaching that point with any degree of confidence required a disciplined approach aimed at collecting data and structuring the problem in sufficient detail to allow for the desired analysis. The following sections highlight the research steps taken.

A review of recent literature provided the baseline upon which to conceptualize the space debris environment system and the approach in modeling that system. Other researchers have developed models simulating the debris environment, which can be categorized by the way they partition the environment for the calculation of debris spatial densities. Whether these researchers defined the densities as depending on altitude and some other parameter or by altitude alone, the resulting collision probabilites did not differ significantly. Although past models aided in conceptualizing and structuring my model, their results did not provide direct points of comparison since most only considered smaller

spacecraft, namely the Space Shuttle, as the satellite-of-interest.

The literature review also highlighted the primary system elements in the space debris environment system. The end result of this research showed how little is known about the actual dynamics of the system. The most significant unknowns include the actual magnitude of the untracked debris population, the rate at which unintentional explosions occur, and the actual dynamics of these and ASAT test explosions.

Systems conceptualization consisted of collecting the information obtained from the literature review and organizing it in such a manner as to be able to understand the relationships between system elements. I used a causal diagram for this purpose. The end result indicated an inherent instability in the space debris environment system where uncontrolled growth the debris population could occur.

Development of the parametric model consisted of selecting those system elements from the conceptual model which I felt could be reasonably represented by computer code and building a model simulating those elements and the relationships between them. I selected the SLAM simulation language and built a discrete-event simulation using Fortran subroutines. Each subroutine corresponded to a separate system

element which SLAM called and scheduled over a 30-year period encompassing initial deployment, growth, and full operational capability of the Space Station.

The verification and validation process attempted to bring credibility to my model and increase the confidence in the analysis of the model's output. Verification was a relatively straight-forward process using coding within each subroutine to trace the changing values of model parameters. Validation was much more difficult due to the unknowns pertaining to the space debris environment system and the fact that the model was making predictions 30 years into the future. The validation process consisted of comparing model output to historical data on those system elements that are understood, comparing cause-effect relationships between certain elements of the parametric model with predicted relationships found in the conceptual model, and comparing collision probability calculations to calculations made by other researchers.

Analysis of model output involved the probability of collision, the number of encounters with debris and the subsequent impact on Space Station fuel requirements, and the effect of launch rates and the number of potentially explodable objects on the collision probability. Using variations of the

baseline model which altered the size of the initial untrackable debris population, analysis of the collision probability showed that at least one collision could occur between 25 and 29 years on the average. The worst case indicated that a collision could occur in as little as 16 years.

Calculation of the number of encounters with debris requiring avoidance maneuvers yielded very large values when using a method involving swept volumes. I also used an alternate calculation involving the Poisson distribution to obtain what would seem to be more realistic values. However, while the second method did result in generally lower values, the results were still large enough to create an impact on Space Station fuel requirements if the spacecraft was required to maneuver away from an object any further away than one kilometer.

Sensitivity analysis on launch rates and the number of potentially explodable objects deposited by launches revealed the overwhelming impact of the second system element on the debris population, and hence on the collision probability. An increasing launch rate, easily predicted based on current trends, combined with efforts to minimize the number of potentially explodable objects produced by launches could lengthen the expected time until a collision occurs to almost

70 years.

Conclusions

Any attempt to model a system depends on what is known or can be reasonably assumed about that system. This research effort is no different. The remaining unknowns concerning the space debris environment are critical in obtaining an accurate assessment of its impact on man's use of space. However, I do not feel that this warrants the labeling of my research effort as unrealistic or trivial. Rather, it merely caveats the results which I believe serve the purpose for which they were intended--to provide an initial indication of the possible severity of debris problem with regard to Space Station operations.

With this in mind, a determination as to the accuracy of my calculations is not what is neeaed. Instead, individuals interested in this field must decide whether my results are optimistic or pessimistic. Should they prove to be optimistic, further research on the debris environment would not be needed. However, efforts to increase the survivability of the Space Station would be.

I feel that my probability of collision calculations are indeed optimistic. The combination of the unknown magnitude of the untracked debris population, the known lethality of these small

particles, and the constant exposure of the Space Station to such an environment over extremely long periods of time points to a very real problem only lightly considered in literature on the Space Station. Should the collision rate prove to be even greater than my calculations indicate, the Space Station will barely have reached system maturity until it is in great danger.

While I feel that my collision probabilities are representative of the actual situation and possibly even a bit optimistic, the debris encounter calculations appear to be unrealistic even for the predicted severity of the environment. Even if the untracked debris could be factored out of the calculations, the numbers would most probably suggest a totally unsurvivable situation. The space debris environment model as designed may be sufficient to calculate realistic collision probabilities, but may not provide enough detail to accurately access the occurrences of encounters requiring avoidance maneuvers.

In argueing for my debris encounter results, I feel that they have some merit until a more convincing calculation method is developed. Unless new results totally discount my calculations, the number of debris encounters indicate that the tracking ability of

ground-based facilities must drastically improve if the Space Station is going to survive without constantly maneuvering and being resupplied with fuel. Even with increased tracking capability, a serious trade-off exists between the financial costs and logistical problems associated with increased resupply rates and the degree of risk acceptable for potential collisions.

The significance of the sensitivity analysis results lies not in the actual numbers obtained but in the realization that acting upon system elements under human control can tremendously lessen the severity of the problem. The use of space can and should not be controlled, but care should be taken in putting only objects in orbit which are functional.

Recommendations

Studies concerning the space debris environment have been conducted as long as the space program has existed. Unfortunately, I noticed the same recommendations being made repeatedly with no apparent orderly progression in the problem-solving process nor any serious consideration by decision-makers of study results. Any general recommendations I make on the problem have been repeated numerous times. However, I feel any recommendations are worth repeating because of their derivation from problems I experienced during my

research effort which proved to be major obstacles. In addition, I have several recommendations specific to my research effort which may enhance the orderly progression of my research.

I only have one general recommendation, but it is by far the most critical issue underlying attempts to study the impact of the debris environment on various spacecraft operations. Solving for the unknowns surrounding the space debris environment is critical to obtaining credible results. While I am sure my literature review did not encompass all of the available research, especially that concerning estimations of explosion dynamics, I do know that other researchers continue to call for more detailed research into the internal workings of the space debris environment system.

Unfortunately, the amount of work I saw remaining in this effort grew as the time left to do it in shrank. While I felt I accomplished my objectives, several aspects of the research effort warrant further work. First, the calculation of debris encounters appears suspect and may require a more detailed structuring of the model to obtain what may prove to be more reasonable results. Suggestions include a conversion of the model from a one- to a two-dimensional model utilizing altitude and

inclination or latitude, and the narrowing of the altitude bands. These changes could give a more accurate assessment of exact debris locations, and therefore more accurate counts of debris within the selected Space Station buffer zone. However, this level of detail would have to be evaluated against the accuracy of predicting future system states. Second, a more in-depth statistical analysis could be conducted using multivariate techniques to analyze system element interactions and impacts with collision probabilities and debris encounters.

I hope the results of this thesis create increased interest in the space debris problem involving the proposed Space Station. The spacecraft's size, cost, operational life, and international and commercial interest warrant such attention.

Appendix A

This appendix presents the space debris environment parametric model.

```
************************************************************************
************************************************************************
*                                                                      *
*                                                                      *
*                                                                      *
*                SPACE DEBRIS ENVIRONMENT MODEL                        *
*                                                                      *
*                     SLAM DISCRETE EVENT                              *
*                                                                      *
*                          DEC 1984                                    *
*                                                                      *
*                                                                      *
*                                                                      *
************************************************************************
************************************************************************
*                                                                      *
************************************************************************
*                                                                      *
*                         MAIN PROGRAM                                 *
*                                                                      *
************************************************************************
*
      program main
      dimension nset(75000)
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common qset(75000)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
```

```
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      equivalence(nset(1),qset(1))
      nnset=75000
      ncrdr=5
      open(8,file='out31',status='old')
      nprnt=6
      ntape=7
      call slam
      stop
      end
*
***********************************************************************
*                                                                     *
*                 EVENT - SCHEDULING SUBROUTINE                       *
*                                                                     *
***********************************************************************
*
      subroutine event(i)
      go to (1,2,3,4,5,6,7,8,9),i
1     call launch
      return
2     call explod
      return
3     call asat
      return
4     call decay
      return
5     call soicol
      return
6     call iocolh
      return
```

```
7      call iocolm
       return
8      call iocoll
       return
9      call check
       return
       end
*
*********************************************************************
*                                                                   *
*                  INITIALIZATION SUBROUTINE                        *
*                                                                   *
*********************************************************************
*
       subroutine intlc
*
*      this subroutine initializes variables
*
       common/scoml/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
      +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
      +tnow,xx(100)
*
       common/ucoml/area,vello,velmd,velhi,deltlo,deltmd,delthi,
      +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
      +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
      +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
      +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
      +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
      +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
      +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
      +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
      +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
      +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
      +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
      +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
       common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
      +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
      +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
      +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
      +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
      +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
      +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
      +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
      +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
      +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
      +   soivsh,numrun,c
       integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
      +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
      +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
      +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
      +   exphin(30),exptn(30),deltlo,deltmd,delthi,
      +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
      +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
      +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
```

```
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      integer i,j
*
*     initialize starting calendar year
*
      cyear=1984
*
*     initialize array elements to zero for first simulation run
*       for only those arrays that collect data over all simulation
*       runs
*
      if(nnrun.eq.1.0) then
       do 20 i=1,30
         tprolo(i)=0.0
         tpromd(i)=0.0
         tprohi(i)=0.0
         sqarlo(i)=0.0
         sqarmd(i)=0.0
         sqarhi(i)=0.0
         meanlo(i)=0.0
         meanmd(i)=0.0
         meanhi(i)=0.0
         varlo(i)=0.0
         varmd(i)=0.0
         varhi(i)=0.0
         sumlow(i)=0
         summed(i)=0
         sumhi(i)=0
         sumtot(i)=0
         sumexp(i)=0
         sumlow(i)=0
         summed(i)=0
         sumhi(i)=0
         sumtot(i)=0
         smexpl(i)=0
         smexpm(i)=0
         smexph(i)=0
         sumexp(i)=0
         smllrt(i)=0
```

```
          smlmrt(i)=0
          smlhrt(i)=0
          smltrt(i)=0
          smalrt(i)=0
          smamrt(i)=0
          smahrt(i)=0
          smatrt(i)=0
          smelrt(i)=0
          smemrt(i)=0
          smehrt(i)=0
          smetrt(i)=0
          smiocl(i)=0
          smiocm(i)=0
          smioch(i)=0
          smioct(i)=0
          smnobl(i)=0
          smnobm(i)=0
          smnobh(i)=0
          mxobl(i)=0
          mxobm(i)=0
          mxobh(i)=0
          year(i)=cyear
          cyear=cyear+1
20      continue
      end if
*
*     initialize all other array elements to zero
*
      do 30  j=1,30
          problo(j)=0.0
          probmd(j)=0.0
          probhi(j)=0.0
          low(j)=0
          med(j)=0
          hi(j)=0
          tot(j)=0
          exptn(j)=0
          explon(j)=0
          expmdn(j)=0
          exphin(j)=0
          alrt(j)=0
          amrt(j)=0
          ahrt(j)=0
          atrt(j)=0
          elrt(j)=0
          emrt(j)=0
          ehrt(j)=0
          etrt(j)=0
          llrt(j)=0
          lmrt(j)=0
          lhrt(j)=0
          ltrt(j)=0
          ioclrt(j)=0
          iocmrt(j)=0
```

```
        iochrt(j)=0
        ioctrt(j)=0
        numobl(j)=0
        numobm(j)=0
        numobh(j)=0
   30 continue
*
*     set number of simulation runs desired
*
      numrun=9.0
*
*     initialize COUNT to start counting weeks at week 1
*
      count=1
*
*     initialize SOI cross-sectional area
*
      area=0.0017958
*
*     initialize variables determining max number of encounters
*       per quarter in each altitude band to zero
*
      mxobl1=0
      mxobl2=0
      mxobl3=0
      mxobl4=0
      mxobm1=0
      mxobm2=0
      mxobm3=0
      mxobm4=0
      mxobh1=0
      mxobh2=0
      mxobh3=0
      mxobh4=0
*
*     set explodable object indicator to show exlodable population
*       is greater than zero
*
      flagex=1
*
*     set average orbital velocities for debris in each altitude
*       band
*
      vello=7.726281
      velmd=7.6130945
      velhi=7.478829
*
*     set altitude band upper boundaries
*
      altlo=400
      altmd=600
      althi=900
```

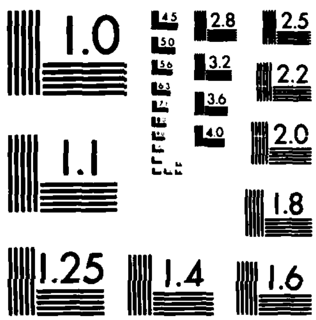
AD-A151 872 ANALYSIS OF SPACE STATION OPERATIONS IN THE SPACE DEBRIS ENVIRONMENT(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 3/3

UNCLASSIFIED B M WAECHTER DEC 84 AFIT/GOR/OS/84D-15 F/G 22/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
*
*     set initial tracked populations
*
      satplo=311
      satpmd=830
      satphi=1452
      satpop=2593
*
*     determine initial small object populations
*
      smallo=0.94*satplo
      smalmd=0.90*satpmd
      smalhi=0.934*satphi
*
*     determine initial large object populations
*
      larglo=satplo-smallo
      largmd=satpmd-smalmd
      larghi=satphi-smalhi
*
*     determine debris populations including untrackable objects
*
      satplo=1244
      satpmd=3320
      satphi=5808
      satpop=10372
*
*     recalculate small object populations to include untrackable
*       debris
*
      smallo=satplo-larglo
      smalmd=satpmd-largmd
      smalhi=satphi-larghi
*
*     set altitude band orbital decay constants
*
      dlo=0.00857
      dmd=0.00444
      dhi=0.0001166
*
*     set CLOCK to correspond to first year of simulated time
*
      clock=tnow+1
*
*     calculate volume swept out per orbit by SOI in each altitude
*       band
*
      soivsl=0.0
      soivsm=(2*3.1415927)*(6378+500)*(314.15927)
      soivsh=0.0
```

```
*
*     calculate proportion of total debris population found in
*       each altitude band
*
      rello=satplo/(satpop*1.0)
      relmd=satpmd/(satpop*1.0)
      relhi=satphi/(satpop*1.0)
*
*     set initial total explodable object population in altitudes of
*       interest
*
      expobj=713
*
*     determine proportion of total explodable object population in
*       each altitude band
*
      explo=expobj*rello
      expmd=expobj*relmd
      exphi=expobj-explo-expmd
*
*     set time in between events
*
      tidcay=1.92308E-02
      tiioc=2.7472571E-03
      tisoic=1.92308E-02
      tiexpl=expon(500/(expobj*1.0),10)
      tiasat=expon(0.5,1)
      if(tiasat.lt.0.33) tiasat=0.33
      if(tiasat.gt.1.0) tiasat=1.0
      tilnch=expon(.0074,8)
      if(tilnch.lt.0.0067) tilnch=0.0067
      if(tilnch.gt.0.0083) tilnch=0.0083
*
*     schedule all events
*
      call schdl(1,tilnch,atrib)
      call schdl(2,tiexpl,atrib)
      call schdl(3,tiasat,atrib)
      call schdl(4,0.015,atrib)
      call schdl(5,8.0151,atrib)
      call schdl(6,0.016,atrib)
      call schdl(7,0.0161,atrib)
      call schdl(8,0.0162,atrib)
      call schdl(9,1.0,atrib)
      return
      end
*
```

```
***********************************************************************
*                                                                     *
*                       ASAT TEST SUBROUTINE                          *
*                                                                     *
***********************************************************************
*
      subroutine asat
*
*     this subroutine deals with the occurrence of debris-depositing
*         asat tests
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
```

```
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      real aaltpb
*
*     schedule next asat test that results in debris deposition
*
      tiasat=expon(0.50,1)
      if(tiasat .lt. 0.33) tiasat=0.33
      if(tiasat .gt. 1.0) tiasat=1.0
      call schdl(3,tiasat,atrib)
*
*     determine altitude band where asat test occurs
*     determine debris added
*
      aaltpb=unfrm(0.0,1.0,2)
      clock=tnow+1
      if(aaltpb .ge. 0.0 .and. aaltpb .le. 0.25) then
          atrib(7)=rnorm(75.0,15.0,3)
          atrib(8)=0.0
          atrib(9)=0.0
          alrt(clock)=alrt(clock)+1
*
      else if(aaltpb .gt. 0.25 .and. aaltpb .le. 0.75) then
          atrib(7)=0.0
          atrib(8)=rnorm(247.0,39.0,3)
          atrib(9)=0.0
          amrt(clock)=amrt(clock)+1
*
      else
          atrib(7)=0.0
          atrib(8)=0.0
          atrib(9)=rnorm(140.0,50.0,3)
          ahrt(clock)=ahrt(clock)+1
*
      end if
*
*     update satellite populations by altitude band
*
      call uf(6)
      return
      end
*
```

```
*********************************************************************
*                                                                   *
*                   ORBITAL DECAY SUBROUTINE                        *
*                                                                   *
*********************************************************************
*
      subroutine decay
*
*     this subroutine deals with the orbital decay of debris through
*         the altitude bands
*
      common/scoml/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucoml/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
```

```
+    smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
+    mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
     real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
+    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
+    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
+    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
+    soivsl,soivsm,soivsh,numrun
     double precision c,colide,rholo,rhomd,rhohi
*
*    schedule next decay update-weekly
*
     call schdl(4,tidcay,atrib)
*
*    update satellite populations subject to decay rates
*
     call uf(7)
*
     return
     end
*
*************************************************************************
*                                                                       *
*                 UNINTENTIONAL EXPLOSION SUBROUTINE                    *
*                                                                       *
*************************************************************************
*
      subroutine explod
*
*     this subroutine deals with unintentional explosions
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
```

```
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      real ealtpb
*
*     schedule next explosion
*
      if(expobj .eq. 0) then
          flagex=0
          go to 1
      else if(flagex.eq.0.and.expobj.gt.0) then
          flagex=1
          tiexpl=expon(500/(expobj*1.0),10)
          go to 21
      else
          tiexpl=expon(500/(expobj*1.0),10)
      end if
      call schdl(2,tiexpl,atrib)
*
*     determine altitude band where explosion occurs
*     determine debris added
*     decrement number of explodable objects by altitude band
*
1     ealtpb=unfrm(0.0,1.0,2)
      clock=tnow+1
*
```

```
      if(ealtpb.ge.0.0.and.ealtpb.le.(explo/(expobj*1.0))) then
          atrib(4)=gama(500.0,140.0,4)
          if(atrib(4).lt.5) atrib(4)=5
          if(atrib(4).gt.15000) atrib(4)=15000
          elrt(clock)=elrt(clock)+1
*
          atrib(5)=0.0
*
          atrib(6)=0.0
*
          explo=explo-1
          satplo=satplo-1
*
      else if(ealtpb.gt.(explo/expobj).and.ealtpb.le.((explo+expmd)/
     +(expobj*1.0)))then
          atrib(4)=0.0
*
          atrib(5)=gama(500.0,140.0,4)
          if(atrib(5).lt.5) atrib(5)=5
          if(atrib(5).gt.15000) atrib(5)=15000
          emrt(clock)=emrt(clock)+1
*
          atrib(6)=0.0
*
          expmd=expmd-1
          satpmd=satpmd-1
*
      else
*
          atrib(4)=0.0
*
          atrib(5)=0.0
*
          atrib(6)=gama(500.0,140.0,4)
          if(atrib(6).lt.5) atrib(6)=5
          if(atrib(6).gt.15000) atrib(6)=15000
          ehrt(clock)=ehrt(clock)+1
*
          exphi=exphi-1
          satphi=satphi-1
*
      end if
*
*     decrement the total number of explodable objects
*
      expobj=expobj-1
      satpop=satpop-1
*
*     update satellite populations by altitude band
*
      call uf(2)
 21   return
      end
*
```

```
************************************************************************
*                                                                      *
*      HIGH ALTITUDE BAND INTER-OBJECT COLLISION SUBROUTINE            *
*                                                                      *
************************************************************************
*
      subroutine iocolh
*
*     this subroutine deals with the collision of two objects, other
*         than the soi, in the high altitude band
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
```

```
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +    soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      real iochpb,sschpb,slchpb,llchpb
*
*     schedule next inter-object collision (high band) update
*         -daily basis
*
      call schdl(6,tiioc,atrib)
*
*     compute probability of inter-object collision (high band)
*     c=(1/volume of high band) * average object area * average
*         relative velocity * time period
*
      c=2.6309424D-12
      colide=(1-(1/exp(satphi*c)))*(satphi/2)
*
*     determine whether collision occurred
*     if collision, calculate collision probabilities for object
*         sizes
*
      iochpb=unfrm(0.0,1.0,2)
      clock=tnow+1
      if(iochpb .ge. 0.0 .and. iochpb .le. colide) then
          iochrt(clock)=iochrt(clock)+1
*
*         calculate probability of high band small-small object
*             collision
*
          sschpb=((smalhi+0.0)/(satphi*1.0))*
     +    ((smalhi-1.0)/((satphi*1.0)-1.0))
*
*         calculate probability of high band small-large object
*             collision
*
          slchpb=(((smalhi+0.0)/(satphi*1.0))*
     +    (larghi/((satphi*1.0)-1.0)))+(((larghi+0.0)/(satphi*1.0))
     +    *(smalhi/((satphi*1.0)-1.0)))
*
*         calculate probability of high band large-large object
*             collision
*
          llchpb=1-sschpb-slchpb
*
*         determine whether collision involved explodable objects
*
          numco=unfrm(0.0,1.0,2)
```

```
          if(numco.le.((exphi/(satphi*1.0))*
     +    ((satphi-exphi)/((satphi*1.0)-1.0))+((satphi-exphi)/
     +    (satphi*1.0))*(exphi/((satphi*1.0)-1.0)))) then
             exphi=exphi-1
             expobj=expobj-1
          else if(numco.le.((exphi/(satphi*1.0))*((exphi-1.0)/
     +    ((satphi*1.0)-1.0)))) then
             exphi=exphi-2
             expobj=expobj-2
          end if
          go to 31
      else
          typeco=99
          go to 10
      end if
*
*     determine type of collision based upon just calculated
*         collision probabilities
*     allocate collision debris to altitude bands
*
31    iochpb=unfrm(0.0,1.0,2)
      if(iochpb .ge. 0.0 .and. iochpb .le. sschpb) then
          atrib(10)=0.0
*
          atrib(11)=0.0
*
          atrib(12)=gama(10.0,5.0,5)
          if(atrib(12).lt.2) atrib(12)=2
          if(atrib(12).gt.50) atrib(12)=50
*
          typeco=0
*
      else if(iochpb .gt. sschpb .and. iochpb .le.
     +    (sschpb+slchpb)) then
          atrib(10)=0.0
*
          atrib(11)=0.0
*
          atrib(12)=gama(50.0,10.0,5)
          if(atrib(12).lt.2) atrib(12)=2
          if(atrib(12).gt.200) atrib(12)=200
*
          typeco=1
*
      else
          atrib(10)=0.0
*
          atrib(11)=0.0
*
          atrib(12)=gama(500.0,140.0,5)
          if(atrib(12).lt.5) atrib(12)=5
          if(atrib(12).gt.15000) atrib(12)=15000
*
          typeco=2
```

```
*
      end if
*
*
*     update satellite population by altitude band
*
      call uf(3)
10    return
      end
*
***********************************************************************
*                                                                     *
*     MEDIUM ALTITUDE BAND INTER-OBJECT COLLISION SUBROUTINE          *
*                                                                     *
***********************************************************************
*
      subroutine iocolm
*
*     this subroutine deals with the collision between two objects,
*         other than the soi, in the medium altitude band
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
```

```
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      real iocmpb,sscmpb,slcmpb,llcmpb
*
*     schedule next inter-object collision (medium band) update
*         -daily basis
*
      call schdl(7,tiioc,atrib)
*
*     compute probability of inter-object collision (medium band)
*     c=(1/volume of medium band) * average object area * average
*         relative velocity * time period
*
      c=5.642614D-12
      colide=(1-(1/exp(satpmd*c)))*(satpmd/2)
*
*     determine whether collision occurred
*     if collision occurred, calculate collision probabilities
*         for object sizes
*
      iocmpb=unfrm(0.0,1.0,2)
      clock=tnow+1
      if(iocmpb .ge. 0.0 .and. iocmpb .le. colide) then
          iocmrt(clock)=iocmrt(clock)+1
*
*         calculate probability of medium band small-small object
*             collision
*
          sscmpb=((smalmd+0.0)/(satpmd*1.0))*
     +    ((smalmd-1.0)/((satpmd*1.0)-1.0))
*
*         calculate probability of medium band small-large object
*             collision
*
          slcmpb=(((smalmd+0.0)/(satpmd*1.0))*
     +    (largmd/((satpmd*1.0)-1.0)))+(((largmd+0.0)/(satpmd*1.0))
     +    *(smalmd/((satpmd*1.0)-1.0)))
```

```
*
*         calculate probability of medium band large-large object
*             collision
*
          llcmpb=1-sscmpb-slcmpb
*
*         determine if collision involved an explodable object
*
          numco=unfrm(0.0,1.0,2)
          if(numco.le.((expmd/(satpmd*1.0))*
     +    ((satpmd-expmd)/((satpmd*1.0)-1.0))+((satpmd-expmd)/
     +    (satpmd*1.0))*(expmd/((satpmd*1.0)-1.0)))) then
             expmd=expmd-1
             expobj=expobj-1
          else if(numco.le.((expmd/(satpmd*1.0))*((expmd-1.0)/
     +    ((satpmd*1.0)-1.0)))) then
             expmd=expmd-2
             expobj=expobj-2
          end if
          go to 41
      else
          typeco=99
          go to 10
      end if
*
*     determine type of collision based upon just calculated
*         collision probabilities
*     allocate collision debris to altitude bands
*
41    iocmpb=unfrm(0.0,1.0,2)
      if(iocmpb .ge. 0.0 .and. iocmpb .le. sscmpb) then
          atrib(13)=0.0
*
          atrib(14)=gama(10.0,5.0,6)
          if(atrib(14).lt.2) atrib(14)=2
          if(atrib(14).gt.50) atrib(14)=50
*
          atrib(15)=0.0
*
          typeco=0
*
      else if(iocmpb .gt. sscmpb .and. iocmpb .le.
     +    (sscmpb+slcmpb)) then
          atrib(13)=0.0
*
          atrib(14)=gama(50.0,10.0,6)
          if(atrib(14).lt.2) atrib(14)=2
          if(atrib(14).gt.200) atrib(14)=200
*
          atrib(15)=0.0
*
          typeco=1
*
```

```
      else
          atrib(13)=0.0
*
          atrib(14)=gama(500.0,140.0,6)
          if(atrib(14).lt.5) atrib(14)=5
          if(atrib(14).gt.15000) atrib(14)=15000
*
          atrib(15)=0.0
*
          typeco=2
*
      end if
*
*     update satellite population by altitude band
*
      call uf(4)
10    return
      end
*
**********************************************************************
*                                                                    *
*     LOW ALTITUDE BAND INTER-OBJECT COLLISION SUBROUTINE            *
*                                                                    *
**********************************************************************
*
      subroutine iocoll
*
*     this subroutine deals with the collision between two objects,
*         other than the soi, in the low altitude band
*
      common/scoml/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucoml/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobml,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
```

```
+    smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
+    smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
+    smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
+    soivsh,numrun,c
     integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
+    objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
+    nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
+    low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
+    exphin(30),exptn(30),deltlo,deltmd,delthi,
+    larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
+    sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
+    explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
+    elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
+    iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
+    numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
+    mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
+    smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
+    smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
+    smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
+    smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
+    mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
     real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
+    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
+    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
+    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
+    soivsl,soivsm,soivsh,numrun
     double precision c,colide,rholo,rhomd,rhohi
     real ioclpb,ssclpb,slclpb,llclpb
*
*    schedule next inter-object collision (low band) update
*        -daily basis
*
     call schdl(8,tiioc,atrib)
*
*    compute probability of inter-object collision (low band)
*    c=(1/volume of low band) * average object area * average
*        relative velocity * time period
*
     c=4.2879685D-12
     colide=(1-(1/exp(satplo*c)))*(satplo/2)
*
*    determine whether collision occurred
*    if collision occurred, calculate collision probabilities
*        for object sizes
*
     ioclpb=unfrm(0.0,1.0,2)
     clock=tnow+1
     if(ioclpb .ge. 0.0 .and. ioclpb .le. colide) then
         ioclrt(clock)=ioclrt(clock)+1
```

```
*
*         calculate probability of low band small-small object
*             collision
*
          ssclpb=((smallo+0.0)/(satplo*1.0))*
     +    ((smallo-1.0)/((satplo*1.0)-1.0))
*
*         calculate probability of low band small-large object
*             collision
*
          slclpb=(((smallo+0.0)/(satplo*1.0))*
     +    (larglo/((satplo*1.0)-1.0)))+(((larglo+0.0)/(satplo*1.0))
     +    *(smallo/((satplo*1.0)-1.0)))
*
*         calculate probability of low band large-large object
*             collision
*
          llclpb=1-ssclpb-slclpb
*
*         determine if collision involved an explodable object
*
          numco=unfrm(0.0,1.0,2)
          if(numco.le.((explo/(satplo*1.0))*
     +    ((satplo-explo)/((satplo*1.0)-1.0))+((satplo-explo)/
     +    (satplo*1.0))*(explo/((satplo*1.0)-1.0)))) then
             explo=explo-1
             expobj=expobj-1
          else if(numco.le.((explo/(satplo*1.0))*((explo-1.0)/
     +    ((satplo*1.0)-1.0)))) then
             explo=explo-2
             expobj=expobj-2
          end if
          go to 51
      else
          typeco=99
          go to 10
      end if
*
*     determine type of collision based upon just calculated
*         collision probabilities
*     allocate collision debris to altitude bands
*
51    ioclpb=unfrm(0.0,1.0,2)
      if(ioclpb .ge. 0.0 .and. ioclpb .le. ssclpb) then
          atrib(16)=gama(10.0,5.0,7)
          if(atrib(16).lt.2) atrib(16)=2
          if(atrib(16).gt.50) atrib(16)=50
*
          atrib(17)=0.0
*
          atrib(18)=0.0
*
          typeco=0
*
```

```
      else if(ioclpb .gt. ssclpb .and. ioclpb .le.
     +(ssclpb+slclpb)) then
          atrib(16)=gama(50.0,10.0,7)
          if(atrib(16).lt.2) atrib(16)=2
          if(atrib(16).gt.200) atrib(16)=200
*
          atrib(17)=0.0
*
          atrib(18)=0.0
*
          typeco=1
*
      else
          atrib(16)=gama(500.0,140.0,7)
          if(atrib(16).lt.5) atrib(16)=5
          if(atrib(16).gt.15000) atrib(16)=15000
*
          atrib(17)=0.0
*
          atrib(18)=0.0
*
          typeco=2
*
      end if
*
*     update satellite population by altitude band
*
      call uf(5)
10    return
      end
*
**********************************************************************
*                                                                    *
*                       SPACE LAUNCH SUBROUTINE                      *
*                                                                    *
**********************************************************************
*
      subroutine launch
*
*     this subroutine generates space launches into the altitude
*       bands of interest
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
```

```
+    rhohi,low(30),med(30),hi(30),tot(30),explon(30),
+    expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
+    larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
+    sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
  common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
+    emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
+    ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
+    numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
+    mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
+    smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
+    smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
+    smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
+    smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
+    smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
+    soivsh,numrun,c
  integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
+    objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
+    nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
+    low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
+    exphin(30),exptn(30),deltlo,deltmd,delthi,
+    larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
+    sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
+    explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
+    elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
+    iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
+    numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
+    mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
+    smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
+    smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
+    smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
+    smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
+    mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
  real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
+    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
+    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
+    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
+    soivsl,soivsm,soivsh,numrun
  double precision c,colide,rholo,rhomd,rhohi
  real laltpb
*
*     schedule next launch
*
      tilnch=expon(.0074,8)
      if (tilnch .lt. .0067) tilnch=.0067
      if (tilnch .gt. .0083) tilnch=.0083
      call schdl(1,tilnch,atrib)
```

```
*
*     determine altitude band launch vehicle enters
*     determine debris and explodable objects added
*
      laltpb=unfrm(0.0,1.0,2)
      clock=tnow+1
      if(laltpb .ge. 0.0 .and. laltpb .le. 0.69) then
          atrib(1)=rnorm(13.0,3.0,9)
          if(atrib(1).lt.9) atrib(1)=9
          if(atrib(1).gt.18) atrib(1)=18
*
          atrib(2)=0.0
*
          atrib(3)=0.0
*
          explo=explo+2
          llrt(clock)=llrt(clock)+1
*
       else if(laltpb .gt. 0.69 .and. laltpb .le. 0.84) then
          atrib(1)=0.0
*
          atrib(2)=rnorm(13.0,3.0,9)
          if(atrib(2).lt.9) atrib(2)=9
          if(atrib(2).gt.18) atrib(2)=18
*
          atrib(3)=0.0
*
          expmd=expmd+2
          lmrt(clock)=lmrt(clock)+1
*
       else
          atrib(1)=0.0
*
          atrib(2)=0.0
*
          atrib(3)=rnorm(13.0,3.0,9)
          if(atrib(3).lt.9) atrib(3)=9
          if(atrib(3).gt.18) atrib(3)=18


*
          exphi=exphi+2
          lhrt(clock)=lhrt(clock)+1
       end if
*
      expobj=expobj+2
      if(flagex.eq.0) call event(2)
*
*     update satellite populations by altitude band
*
      call uf(1)
      return
      end
*
```

```
***********************************************************************
*                                                                     *
*        SOI COLLISION PROBABILITY CALCULATION SUBROUTINE             *
*                                                                     *
***********************************************************************
*
      subroutine soicol
*
*     this subroutine deals with the calculation of collision between a
*        satellite of interest (SOI) and another object
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     :   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
```

```
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
*
*     schedule next satellite of interest collision probability
*        calculation
*
      call schdl(5,tisoic,atrib)
*
*     calculate soi collision probability by altitude band
*
      call uf(8)
      return
      end
*
*******************************************************************
*                                                                 *
*                    USER FUNCTION SUBROUTINE                     *
*                                                                 *
*******************************************************************
*
      subroutine uf(u)
*
*     this subroutine produces the desired user function inputs
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
```

```
+    smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
+    smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
+    smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
+    smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
+    soivsh,numrun,c
  integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
+    objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
+    nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
+    low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
+    exphin(30),exptn(30),deltlo,deltmd,delthi,
+    larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
+    sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
+    explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
+    elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
+    iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
+    numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
+    mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
+    smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
+    smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
+    smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
+    smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
+    mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
  real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
+    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
+    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
+    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
+    soivsl,soivsm,soivsh,numrun
  double precision c,colide,rholo,rhomd,rhohi
  integer u
*
      go to (1,2,3,4,5,6,7,8),u
*
*update all satpop's for launches
*
1     objlo=atrib(1)
      objmd=atrib(2)
      objhi=atrib(3)
      satplo=satplo+objlo
      satpmd=satpmd+objmd
      satphi=satphi+objhi
      smallo=smallo+(0.9*objlo)
      smalmd=smalmd+(0.9*objmd)
      smalhi=smalhi+(0.9*objhi)
      satpop=satplo+satpmd+satphi
      larglo=satplo-smallo
      largmd=satpmd-smalmd
      larghi=satphi-smalhi
      return
```

```
*
*update satpop's for explosions
*
2     nexplo=atrib(4)
      nexpmd=atrib(5)
      nexphi=atrib(6)
      satplo=satplo+nexplo
      satpmd=satpmd+nexpmd
      satphi=satphi+nexphi
      satpop=satplo+satpmd+satphi
      smallo=smallo+(0.95*nexplo)
      smalmd=smalmd+(0.95*nexpmd)
      smalhi=smalhi+(0.95*nexphi)
      larglo=satplo-smallo
      largmd=satpmd-smalmd
      larghi=satphi-smalhi
      return
*
*update satpop,s for a collision (high band)
*
3     cobjlo=atrib(10)
      cobjmd=atrib(11)
      cobjhi=atrib(12)
      if (typeco.eq.0) then
         smallo=smallo+cobjlo
         smalmd=smalmd+cobjmd
         smalhi=smalhi+cobjhi-2
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd
         satphi=satphi+cobjhi-2
         satpop=satplo+satpmd+satphi
      else if (typeco.eq.1) then
         smallo=smallo+(0.9*cobjlo)
         smalmd=smalmd+(0.9*cobjmd)
         smalhi=smalhi+(0.9*cobjhi)-1
         larglo=satplo-smallo
         largmd=satpmd-smalmd
         larghi=satphi-smalhi-1
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd
         satphi=satphi+cobjhi-2
         satpop=satplo+satpmd+satphi
      else if(typeco.eq.2) then
         smallo=smallo+(0.8*cobjlo)
         smalmd=smalmd+(0.8*cobjmd)
         smalhi=smalhi+(0.8*cobjhi)
         larglo=satplo-smallo
         largmd=satpmd-smalmd
         larghi=satphi-smalhi-2
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd
         satphi=satphi+cobjhi-2
         satpop=satplo+satpmd+satphi
```

```
      else
         cobjlo=0
         cobjmd=0
         cobjhi=0
      end if
      return
*
*update satpop,s for a collision (medium band)
*
4     cobjlo=atrib(13)
      cobjmd=atrib(14)
      cobjhi=atrib(15)
      if (typeco.eq.0) then
         smallo=smallo+cobjlo
         smalmd=smalmd+cobjmd-2
         smalhi=smalhi+cobjhi
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd-2
         satphi=satphi+cobjhi
         satpop=satplo+satpmd+satphi
      else if (typeco.eq.1) then
         smallo=smallo+(0.9*cobjlo)
         smalmd=smalmd+(0.9*cobjmd)-1
         smalhi=smalhi+(0.9*cobjhi)
         larglo=satplo-smallo
         largmd=satpmd-smalmd-1
         larghi=satphi-smalhi
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd-2
         satphi=satphi+cobjhi
         satpop=satplo+satpmd+satphi
      else if(typeco.eq.2) then
         smallo=smallo+(0.8*cobjlo)
         smalmd=smalmd+(0.8*cobjmd)
         smalhi=smalhi+(0.8*cobjhi)
         larglo=satplo-smallo
         largmd=satpmd-smalmd-2
         larghi=satphi-smalhi
         satplo=satplo+cobjlo
         satpmd=satpmd+cobjmd-2
         satphi=satphi+cobjhi
         satpop=satplo+satpmd+satphi
      else
         cobjlo=0
         cobjmd=0
         cobjhi=0
      end if
      return
```

```
*
*update satpop,s for a collision (low band)
*
5      cobjlo=atrib(16)
       cobjmd=atrib(17)
       cobjhi=atrib(18)
       if (typeco.eq.0) then
          smallo=smallo+cobjlo-2
          smalmd=smalmd+cobjmd
          smalhi=smalhi+cobjhi
          satplo=satplo+cobjlo-2
          satpmd=satpmd+cobjmd
          satphi=satphi+cobjhi
          satpop=satplo+satpmd+satphi
       else if (typeco.eq.1) then
          smallo=smallo+(0.9*cobjlo)-1
          smalmd=smalmd+(0.9*cobjmd)
          smalhi=smalhi+(0.9*cobjhi)
          larglo=satplo-smallo-1
          largmd=satpmd-smalmd
          larghi=satphi-smalhi
          satplo=satplo+cobjlo-2
          satpmd=satpmd+cobjmd
          satphi=satphi+cobjhi
          satpop=satplo+satpmd+satphi
       else if(typeco.eq.2) then
          smallo=smallo+(0.8*cobjlo)
          smalmd=smalmd+(0.8*cobjmd)
          smalhi=smalhi+(0.8*cobjhi)
          larglo=satplo-smallo-2
          largmd=satpmd-smalmd
          larghi=satphi-smalhi
          satplo=satplo+cobjlo-2
          satpmd=satpmd+cobjmd
          satphi=satphi+cobjhi
          satpop=satplo+satpmd+satphi
       else
          cobjlo=0
          cobjmd=0
          cobjhi=0
       end if
       return
*
*update satpop,s for asat tests
*
6      asatlo=atrib(7)
       asatmd=atrib(8)
       asathi=atrib(9)
       satplo=satplo+asatlo
       satpmd=satpmd+asatmd
       satphi=satphi+asathi
       satpop=satplo+satpmd+satphi
       smallo=smallo+(0.98*asatlo)
       smalmd=smalmd+(0.98*asatmd)
```

```
      smalhi=smalhi+(0.98*asathi)
      larglo=satplo-smallo
      largmd=satpmd-smalmd
      larghi=satphi-smalhi
      return
*
*update all satpop"s for decay
*
7     satplo=satplo+(dmd*satpmd)-(dlo*satplo)
      satpmd=satpmd+(dhi*satphi)-(dmd*satpmd)
      satphi=satphi-(dhi*satphi)
      explo=explo+(dmd*expmd)-(dlo*explo)
      expmd=expmd+(dhi*exphi)-(dmd*expmd)
      exphi=exphi-(dhi*exphi)
      expobj=explo+expmd+exphi
      satpop=satplo+satpmd+satphi
      smallo=smallo+(dmd*smalmd)-(dlo*smallo)
      smalmd=smalmd+(dhi*smalhi)-(dmd*smalmd)
      smalhi=smalhi-(dhi*smalhi)
      larglo=satplo-smallo
      largmd=satpmd-smalmd
      larghi=satphi-smalhi
*
*     sum debris and explodable object populations by altitude band
*
      low(clock)=low(clock)+satplo
      med(clock)=med(clock)+satpmd
      hi(clock)=hi(clock)+satphi
      tot(clock)=tot(clock)+satpop
      explon(clock)=explon(clock)+explo
      expmdn(clock)=expmdn(clock)+expmd
      exphin(clock)=exphin(clock)+exphi
      exptn(clock)=exptn(clock)+expobj
      return
*
*compute the weekly probability of collision for the satellite of
*  interest
*
8     rholo=satplo*8.9286D-12
      rhomd=satpmd*8.4034D-12
      rhohi=satphi*5.2219D-12
      deltlo=0
      deltmd=31536000
      delthi=0
      clock=tnow+1
      if(tnow.ge.8.0.and.tnow.le.17.0) area=area+0.0000004
*
*     calculate probability of collision
*
      problo(clock)=problo(clock)+(1-(1/exp(rholo*area*
     +(0.6*vello)*deltlo)))
      probmd(clock)=probmd(clock)+(1-(1/exp(rhomd*area*
     +(0.6*7.6126922)*deltmd)))
      probhi(clock)=probhi(clock)+(1-(1/exp(rhohi*area*
```

```
     +(0.6*velhi)*delthi)))
*
*     calculate trackable debris encountered that requires maneuvering
*
      numobl(clock)=0
      numobm(clock)=numobm(clock)+soivsm*106.53885*rhomd
      numobh(clock)=0
*
      if(count.eq.13) mxobm1=numobm(clock)
      if(count.eq.26) mxobm2=numobm(clock)-mxobm1
      if(count.eq.39) mxobm3=numobm(clock)-mxobm2-mxobm1
      if(count.eq.52) then
        mxobm4=numobm(clock)-mxobm3-mxobm2-mxobm1
        count=0
      end if
      count=count+1
      if(mxobm1.lt.0) mxobm1=0
      if(mxobm2.lt.0) mxobm2=0
      if(mxobm3.lt.0) mxobm3=0
      if(mxobm4.lt.0) mxobm4=0
*
      return
      end
*
**********************************************************************
*                                                                    *
*              SYSTEM PARAMETER CHECK SUBROUTINE                     *
*                                                                    *
**********************************************************************
*
      subroutine check
*
*     this subroutine computes average debris population, explodable
*        object population, and collision probability per year
*     -yearly basis
*     sums populations in respective altitude bands for use in
*        computing average populations sizes per year
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,deltni,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
```

```
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
     common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
+    emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
+    ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
+    numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobml,
+    mxobm2,mxobm3,mxobm4,mxobhl,mxobh2,mxobh3,mxobh4,smexpl(30),
+    smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
+    smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
+    smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
+    smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
+    smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
+    soivsh,numrun,c
     integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
+    objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
+    nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
+    low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
+    exphin(30),exptn(30),deltlo,deltmd,delthi,
+    larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
+    sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
+    explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
+    elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
+    iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
+    numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
+    mxobml,mxobm2,mxobm3,mxobm4,mxobhl,mxobh2,mxobh3,mxobh4,
+    smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
+    smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
+    smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
+    smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
+    mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
     real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
+    tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
+    sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
+    varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
+    tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
+    soivsl,soivsm,soivsh,numrun
     double precision c,colide,rholo,rhomd,rhohi
     integer max
*
*    schedule next check - yearly basis
*
     call schdl(9,1.0,atrib)
*
*    compute average populations, average collision probability
*
     clock=tnow
     low(clock)=low(clock)/52.0
     med(clock)=med(clock)/52.0
     hi(clock)=hi(clock)/52.0
     tot(clock)=tot(clock)/52.0
     explon(clock)=explon(clock)/52.0
     expmdn(clock)=expmdn(clock)/52.0
     exphin(clock)=exphin(clock)/52.0
     exptn(clock)=exptn(clock)/52.0
     problo(clock)=problo(clock)/52.0
```

```
      probmd(clock)=probmd(clock)/52.0
      probhi(clock)=probhi(clock)/52.0
      ltrt(clock)=llrt(clock)+lmrt(clock)+lhrt(clock)
      atrt(clock)=alrt(clock)+amrt(clock)+ahrt(clock)
      etrt(clock)=elrt(clock)+emrt(clock)+ehrt(clock)
      ioctrt(clock)=ioclrt(clock)+iocmrt(clock)+iochrt(clock)
*
*     determine 90-day period with max number of encounters for
*       particular year
*
      if(mxobm1.gt.mxobm2) then
         max=mxobm1
      else
         max=mxobm2
      end if
      if(max.lt.mxobm3) then
         max=mxobm3
         if(mxobm3.lt.mxobm4) max=mxobm4
      else if(max.lt.mxobm4) then
         max=mxobm4
      end if
      if(max.gt.mxobm(clock)) mxobm(clock)=max
*
      return
      end
*
*********************************************************************
*                                                                   *
*                        OUTPUT SUBROUTINE                          *
*                                                                   *
*********************************************************************
*
      subroutine otput
*
*     this subroutine presents collision probabilities,number of
*       of encounters with debris, maximum numbers of encounters
*       per 90-day basis, number of launches per year, number of
*       explosions per year, number of i/o collisions per year
*
      common/scom1/atrib(100),dd(100),ddl(100),dtnow,ii,mfa,mstop,
     +nclnr,ncrdr,nprnt,nnrun,nnset,ntape,ss(100),ssl(100),tnext,
     +tnow,xx(100)
*
      common/ucom1/area,vello,velmd,velhi,deltlo,deltmd,delthi,
     +   altlo,altmd,althi,numco,satpop,satplo,satpmd,satphi,
     +   expobj,dlo,dmd,dhi,nexplo,nexpmd,nexphi,colide,
     +   objlo,objmd,objhi,cobjlo,cobjmd,cobjhi,rello,relmd,relhi,
     +   asatlo,asatmd,asathi,cyear,tprolo(30),tpromd(30),
     +   tprohi(30),sqarlo(30),sqarmd(30),sqarhi(30),meanlo(30),
     +   meanmd(30),meanhi(30),varlo(30),varmd(30),varhi(30),
     +   problo(30),probmd(30),probhi(30),year(30),rholo,rhomd,
     +   rhohi,low(30),med(30),hi(30),tot(30),explon(30),
     +   expmdn(30),exphin(30),exptn(30),smallo,smalmd,smalhi,
     +   larglo,largmd,larghi,typeco,clock,explo,expmd,exphi,
```

```
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,count,flagex
      common/ucom2/alrt(30),amrt(30),ahrt(30),atrt(30),elrt(30),
     +   emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),iochrt(30),
     +   ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),numobl(30),
     +   numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,mxobm1,
     +   mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,smexpl(30),
     +   smexpm(30),smexph(30),smllrt(30),smlmrt(30),smlhrt(30),
     +   smltrt(30),smalrt(30),smamrt(30),smahrt(30),smatrt(30),
     +   smelrt(30),smemrt(30),smehrt(30),smetrt(30),smiocl(30),
     +   smiocm(30),smioch(30),smioct(30),smnobl(30),smnobm(30),
     +   smnobh(30),mxobl(30),mxobm(30),mxobh(30),soivsl,soivsm,
     +   soivsh,numrun,c
      integer altlo,altmd,althi,satpop,satplo,satpmd,satphi,
     +   objlo,objmd,objhi,asatlo,asatmd,asathi,nexplo,nexpmd,
     +   nexphi,cobjlo,cobjmd,cobjhi,year(30),cyear,expobj,
     +   low(30),med(30),hi(30),tot(30),explon(30),expmdn(30),
     +   exphin(30),exptn(30),deltlo,deltmd,delthi,
     +   larglo,largmd,larghi,typeco,clock,smallo,smalmd,smalhi,
     +   sumlow(30),summed(30),sumhi(30),sumtot(30),sumexp(30),
     +   explo,expmd,exphi,count,alrt(30),amrt(30),ahrt(30),atrt(30),
     +   elrt(30),emrt(30),ehrt(30),etrt(30),ioclrt(30),iocmrt(30),
     +   iochrt(30),ioctrt(30),llrt(30),lmrt(30),lhrt(30),ltrt(30),
     +   numobl(30),numobm(30),numobh(30),mxobl1,mxobl2,mxobl3,mxobl4,
     +   mxobm1,mxobm2,mxobm3,mxobm4,mxobh1,mxobh2,mxobh3,mxobh4,
     +   smexpl(30),smexpm(30),smexph(30),smllrt(30),smlmrt(30),
     +   smlhrt(30),smltrt(30),smalrt(30),smamrt(30),smahrt(30),
     +   smatrt(30),smelrt(30),smemrt(30),smehrt(30),smetrt(30),
     +   smiocl(30),smiocm(30),smioch(30),smioct(30),mxobl(30),
     +   mxobm(30),mxobh(30),smnobl(30),smnobm(30),smnobh(30),flagex
      real area,vello,velmd,velhi,problo(30),probmd(30),probhi(30),
     +   tprolo(30),tpromd(30),tprohi(30),sqarlo(30),sqarmd(30),
     +   sqarhi(30),meanlo(30),meanmd(30),meanhi(30),dlo,dmd,dhi,
     +   varlo(30),varmd(30),varhi(30),rello,relmd,relhi,
     +   tilnch,tiexpl,tiasat,tidcay,tisoic,tiioc,numco,
     +   soivsl,soivsm,soivsh,numrun
      double precision c,colide,rholo,rhomd,rhohi
      integer avglow(30),avgmed(30),avghi(30),avgtot(30),
     +   avexpl(30),avexpm(30),avexph(30),avexpt(30),avllrt(30),
     +   avlmrt(30),avlhrt(30),avltrt(30)
      real avalrt(30),avamrt(30),avahrt(30),avatrt(30),avelrt(30),
     +   avemrt(30),avehrt(30),avetrt(30),aviocl(30),aviocm(30),
     +   avioch(30),avioct(30),avnobl(30),avnobm(30),avnobh(30)
*
*     sum up debris and explodable object populations and launch,asat
*       test,explosion, and i/o collision rates over number of
*       simulation runs
*
      do 10 i=1,30
         sumlow(i)=sumlow(i)+low(i)
         summed(i)=summed(i)+med(i)
         sumhi(i)=sumhi(i)+hi(i)
         sumtot(i)=sumtot(i)+tot(i)
         smexpl(i)=smexpl(i)+explon(i)
```

```
          smexpm(i)=smexpm(i)+expadn(i)
          smexph(i)=smexph(i)+exphin(i)
          sumexp(i)=sumexp(i)+exptm(i)
          smllrt(i)=smllrt(i)+llrt(i)
          smlmrt(i)=smlmrt(i)+lmrt(i)
          smlhrt(i)=smlhrt(i)+lhrt(i)
          smltrt(i)=smltrt(i)+ltrt(i)
          smalrt(i)=smalrt(i)+alrt(i)
          smamrt(i)=smamrt(i)+amrt(i)
          smahrt(i)=smahrt(i)+ahrt(i)
          smatrt(i)=smatrt(i)+atrt(i)
          smelrt(i)=smelrt(i)+elrt(i)
          smemrt(i)=smemrt(i)+emrt(i)
          smehrt(i)=smehrt(i)+ehrt(i)
          smetrt(i)=smetrt(i)+etrt(i)
          smiocl(i)=smiocl(i)+ioclrt(i)
          smiocm(i)=smiocm(i)+iocmrt(i)
          smioch(i)=smioch(i)+iochrt(i)
          smioct(i)=smioct(i)+ioctrt(i)
          smnobl(i)=smnobl(i)+numobl(i)
          smnobm(i)=smnobm(i)+numobm(i)
          smnobh(i)=smnobh(i)+numobh(i)
          sqarlo(i)=sqarlo(i)+(problo(i)*problo(i))
          sqarmd(i)=sqarmd(i)+(probmd(i)*probmd(i))
          sqarhi(i)=sqarhi(i)+(probhi(i)*probhi(i))
          tprolo(i)=tprolo(i)+problo(i)
          tpromd(i)=tpromd(i)+probmd(i)
          tprohi(i)=tprohi(i)+probhi(i)
10    continue
*
*     compute average collision probabilities, debris and explodable
*       object populations, launch, asat test, explosion, i/o
*       collision, and encounter rates
*
      if(nnrun.eq.numrun) then
      do 20 j=1,30
          avglow(j)=sumlow(j)/numrun
          avgmed(j)=summed(j)/numrun
          avghi(j)=sumhi(j)/numrun
          avgtot(j)=sumtot(j)/numrun
          avexpl(j)=smexpl(j)/numrun
          avexpm(j)=smexpm(j)/numrun
          avexph(j)=smexph(j)/numrun
          avexpt(j)=sumexp(j)/numrun
          avllrt(j)=smllrt(j)/numrun
          avlmrt(j)=smlmrt(j)/numrun
          avlhrt(j)=smlhrt(j)/numrun
          avltrt(j)=smltrt(j)/numrun
          avalrt(j)=smalrt(j)/numrun
          avamrt(j)=smamrt(j)/numrun
          avahrt(j)=smahrt(j)/numrun
          avatrt(j)=smatrt(j)/numrun
          avelrt(j)=smelrt(j)/numrun
          avemrt(j)=smemrt(j)/numrun
```

```
      avehrt(j)=smehrt(j)/numrun
      avetrt(j)=smetrt(j)/numrun
      aviocl(j)=smiocl(j)/numrun
      aviocm(j)=smiocm(j)/numrun
      avioch(j)=smioch(j)/numrun
      avioct(j)=smioct(j)/numrun
      avnobl(j)=smnobl(j)/numrun
      avnobm(j)=smnobm(j)/numrun
      avnobh(j)=smnobh(j)/numrun
      meanlo(j)=tprolo(j)/numrun
      meanmd(j)=tpromd(j)/numrun
      meanhi(j)=tprohi(j)/numrun
      varlo(j)=(sqarlo(j)-((tprolo(j)*tprolo(j))/numrun))/(numrun-1)
      varmd(j)=(sqarmd(j)-((tpromd(j)*tpromd(j))/numrun))/(numrun-1)
      varhi(j)=(sqarhi(j)-((tprohi(j)*tprohi(j))/numrun))/(numrun-1)
20    continue
*
      write(unit=8,fmt=1)
1     format(//,13X,'SPACE DEBRIS ENVIRONMENT MODEL')
      write(unit=8,fmt=2)
2     format(//,19X,'LOW ALTITUDE BAND')
      write(unit=8,fmt=3)
3     format(//,7X,'probability',33X,'max enc')
      write(unit=8,fmt=4)
4     format(' ','year   collision  variance   # encounters   per qtr')
      write(unit=8,fmt=5)
5     format(' ',' ')
*
      do 30 k=1,30
         write(unit=8,fmt=6)year(k),meanlo(k),varlo(k),avnobl(k),
     +   mxobl(k)
6        format(' ',I4,4X,F9.8,4X,F9.8,F11.2,I11)
30    continue
*
      write(unit=8,fmt=7)
7     format(//,'year  # launches  % total  #explosions  # i/o coll.s
     +       # asat tests')
      write(unit=8,fmt=8)
8     format(' ',' ')
*
      do 40 m=1,30
         write(unit=8,fmt=9)year(m),avllrt(m),avllrt(m)/
     +   (avltrt(m)*1.0),avelrt(m),aviocl(m),avalrt(m)
9        format(' ',I4,I9,F11.4,F12.2,F13.2,F18.2)
40    continue
*
      write(unit=8,fmt=11)
11    format(//,'      avg debris           avg exp')
      write(unit=8,fmt=12)
12    format(' ','year  pop size  % total  pop size  % total')
      write(unit=8,fmt=13)
13    format(' ',' ')
*
```

```
      do 50 n=1,30
         write(unit=8,fmt=14)year(n),avglow(n),avglow(n)/
     +   (avgtot(n)*1.0),avexpl(n),avexpl(n)/(avexpt(n)*1.0)
14       format(' ',I4,I8,F11.4,I8,F11.4)
50    continue
*
*
      write(unit=8,fmt=15)
15    format(//,16X,'MEDIUM ALTITUDE BAND')
      write(unit=8,fmt=16)
16    format(//,7X,'probability',33X,'max enc')
      write(unit=8,fmt=17)
17    format(' ','year   collision  variance   # encounters   per qtr')
      write(unit=8,fmt=18)
18    format(' ',' ')
*
      do 60 p=1,30
         write(unit=8,fmt=19)year(p),meanmd(p),varmd(p),avnobm(p),
     +   mxobm(p)
19       format(' ',I4,4X,F9.8,4X,F9.8,F11.2,I11)
60    continue
*
      write(unit=8,fmt=21)
21    format(//,'year  # launches  % total  #explosions  # i/o coll.s
     +       # asat tests')
      write(unit=8,fmt=22)
22    format(' ',' ')
*
      do 70 q=1,30
         write(unit=8,fmt=23)year(q),avlmrt(q),avlmrt(q)/
     +   (avltrt(q)*1.0),avemrt(q),aviocm(q),avamrt(q)
23       format(' ',I4,I9,F11.4,F12.2,F13.2,F18.2)
70    continue
*
      write(unit=8,fmt=24)
24    format(//,'      avg debris             avg exp')
      write(unit=8,fmt=26)
26    format(' ','year  pop size  % total  pop size  % total')
      write(unit=8,fmt=27)
27    format(' ',' ')
*
      do 80 r=1,30
         write(unit=8,fmt=28)year(r),avgmed(r),avgmed(r)/
     +   (avgtot(r)*1.0),avexpm(r),avexpm(r)/(avexpt(r)*1.0)
28       format(' ',I4,I8,F11.4,I8,F11.4)
80    continue
*
*
      write(unit=8,fmt=29)
29    format(//,18X,'HIGH ALTITUDE BAND')
      write(unit=8,fmt=31)
31    format(//,7X,'probability',33X,'max enc')
      write(unit=8,fmt=32)
32    format(' ','year   collision  variance   # encounters   per qtr')
```

```
      write(unit=8,fmt=33)
33    format(' ',' ')
*
      do 90 s=1,30
         write(unit=8,fmt=34)year(s),meanhi(s),varhi(s),avnobh(s),
     +   mxobh(s)
34       format(' ',I4,4X,F9.8,4X,F9.8,F11.2,I11)
90    continue
*
      write(unit=8,fmt=36)
36    format(//,'year  # launches  % total  #explosions  # i/o coll.s
     +       # asat tests')
      write(unit=8,fmt=37)
37    format(' ',' ')
*
      do 100 t=1,30
         write(unit=8,fmt=38)year(t),avlhrt(t),avlhrt(t)/
     +   (avltrt(t)*1.0),avehrt(t),avioch(t),avahrt(t)
38       format(' ',I4,I9,F11.4,F12.2,F13.2,F18.2)
100   continue
*
      write(unit=8,fmt=39)
39    format(//,'       avg debris            avg exp')
      write(unit=8,fmt=41)
41    format(' ','year  pop size  % total  pop size  % total')
      write(unit=8,fmt=42)
42    format(' ',' ')
*
      do 110 v=1,30
         write(unit=8,fmt=43)year(v),avghi(v),avghi(v)/
     +   (avgtot(v)*1.0),avexph(v),avexph(v)/(avexpt(v)*1.0)
43       format(' ',I4,I8,F11.4,I8,F11.4)
110   continue
*
      write(unit=8,fmt=44)
44    format(//,'                      OVERALL RESULTS')
      write(unit=8,fmt=46)
46    format(//,'        tot      tot')
      write(unit=8,fmt=47)
47    format(' ','year     pop  explod pop')
      write(unit=8,fmt=48)
48    format(' ',' ')
*
      do 120 w=1,30
         write(unit=8,fmt=49)year(w),avgtot(w),avexpt(w)
49       format(' ',I4,I8,I9)
120   continue
*
      write(unit=8,fmt=51)
51    format(//,'                                              # i/o ')
      write(unit=8,fmt=52)
52    format(//,'year  # launches  # explosions  # asat tests   coll')
*
```

```
      do 130 x=1,30
         write(unit=8,fmt=53)year(x),avltrt(x),avetrt(x),avatrt(x),
     +   avioct(x)
53       format(' ',I4,I9,F14.2,F14.2,F16.2)
130   continue
*
      end if
      return
      end
```

Appendix B

This appendix presents definitions of all user-defined and SLAM-provided variables used in the parametric model. Variables are classified as user-defined discrete variables, user-defined arrays, and SLAM-defined variables. The variables are listed alphabetically, although variables that perform the same function but only differ as to the altitude band they represent are listed together irrespective of alphabetical order.

User-Defined Discrete Variables

ALTLO - upper boundary of low altitude band (km)

ALTMD - upper boundary of medium altitude band (km)

ALTHI - upper boundary of high altitude band (km)

AREA - satellite of interest (SOI) cross-sectional area (km^2)

ASATLO - quantity of debris added to the low altitude band from an ASAT test

ASATMD - quantity of debris added to the medium altitude band from an ASAT test

ASATHI - quantity of debris added to the high altitude band from an ASAT test

C - represents [(1/volume of respective altitude band) x average object area in respective altitude band x average relative velocity between objects in respective altitude band x number of seconds in one day] for an average object in the respective altitude band; used in the calculation of the inter-object collision probability in that altitude band

CLOCK - tracks with the current simulation time to assign values to arrays for the appropriate year

COBJLO - quantity of debris added to the low altitude band from an inter-object collision

COBJMD - quantity of debris added to the medium altitude band from an inter-object collision

COBJHI - quantity of debris added to the high altitude band from and inter-object collision

COLIDE - probability that an inter-object collision will occur

COUNT - keeps track of number of weeks in current year; used in determination of the maximum debris encounters per quarter

CYEAR - calendar year corresponding to the simulation time

DELTLO - amount of time per year the satellite of interest spends in the low altitude band (sec)

DELTMD - amount of time per year the satellite of interest spends in the medium altitude band (sec)

DLO - low altitude band decay constant; the average percentage of low altitude objects that decay out of that band in one week

DMD - medium altitude band decay constant; the average percentage of medium altitude objects that decay out of that band in one week

DHI - high altitude band decay constant; the average percentage of high altitude objects that decay out of that band in one week

EXPLO - potentially explodable object population in the low altitude band

EXPMD - potentially explodable object population in the medium altitude band

EXPHI - potentially explodable object population in the high altitude band

EXPOBJ - total potentially explodable object population

FLAGEX - indicates when total potentially explodable population equals zero

LARGLO - large object (greater than 1 m^2 average radar cross section) population in the low altitude band

LARGMD - large object (greater than 1 m^2 average radar cross section) population in the medium altitude band

LARGHI - large object (greater than 1 m^2 average radar cross section) population in the high altitude band

MXOBL1 - the maximum number of objects encountered by the satellite of interest in the low altitude band in the first quarter of a given year

MXOBL2 - the maximum number of objects encountered by the satellite of interest in the low altitude band in the second quarter of a given year

MXOBL3 - the maximum number of objects encountered by the satellite of interest in the low altitude band in the third quarter of a given year

MXOBL4 - the maximum number of objects encountered by the satellite of interest in the low altitude band in the fourth quarter of a given year

MXOBM1 - the maximum number of objects encountered by the satellite of interest in the medium altitude band in the first quarter of a given year

MXOBM2 - the maximum number of objects encountered by the satellite of interest in the medium altitude band in the second quarter of a given year

MXOBM3 - the maximum number of objects encountered by the satellite of interest in the medium altitude band in the third quarter of a given year

MXOBM4 - the maximum number of objects encountered by the satellite of interest in the medium altitude band in the fourth quarter of a given year

MXOBH1 - the maximum number of objects encountered by the satellite of interest in the high altitude band in the first quarter of a given year

MXOBH2 - the maximum number of objects encountered by the satellite of interest in the high altitude band in the second quarter of a given year

MXOBH3 - the maximum number of objects encountered by the satellite of interest in the high altitude band in the third quarter of a given year

MXOBH4 - the maximum number of objects encountered by the satellite of interest in the high altitude band in the fourth quarter of a given year

NEXPLO - quantity of debris added to the low altitude band from an unintentional explosion

NEXPMD - quantity of debris added to the medium altitude band from an unintentional explosion

NEXPHI - quantity of debris added to the high altitude band from an unintentional explosion

NUMCO - random variate from a uniform (0,1) distribution used to determine if an inter-object collision involved one or two explodable objects

NUMRUN - desired number of simulation runs

OBJLO - quantity of debris added to the low altitude band from a space launch

OBJMD - quantity of debris added to the medium altitude band from a space launch

OBJHI - quantity of debris added to the high altitude band from a space launch

RELLO - proportion of the total space object population present in the low altitude band

RELMD - proportion of the total space object population present in the medium altitude band

RELHI - proportion of the total space object population present in the high altitude band

RHOLO - spatial density of objects in the low altitude band (# objects/km^3)

RHOMD - spatial density of objects in the medium altitude band (# objects/km^3)

RHOHI - spatial density of objects in the high altitude band (# objects/km^3)

SATPLO - space object population in the low altitude band

SATPMD - space object population in the medium altitude band

SATPHI - space object population in the high altitude band

SATPOP - total space object population

SMALLO - small object (less than 1 m^2 average radar cross section) population in the low altitude band

SMALMD - small object (less than 1 m^2 average radar cross section) population in the medium altitude band

SMALHI - small object (less than 1 m^2 average radar cross section) population in the high altitude band

SOIVSL - volume of space swept out in one orbit of the satellite of interest in the low altitude band; calculated as 2 x SOI orbital altitude as measured from the earth's center x selected cross-sectional area greater than or equal to the SOI (km^3/orbit)

SOIVSM - volume of space swept out in one orbit of the satellite of interest in the medium altitude band; calculated as 2 x SOI orbital altitude as measured from the earth's center x selected cross-sectional area greater than or equal to the SOI (km^3/orbit)

SOIVSH - volume of space swept out in one orbit of the satellite of interest in the high altitude band; calculated as 2 x SOI orbital altitude as measured from the earth's center x selected cross-sectional area greater than or equal to the SOI (km^3/orbit)

TIASAT - interval of time between ASAT tests (sec)

TIDCAY - interval of time between updates of the space object population due to orbital decay (sec)

TIEXPL - interval of time between unintentional explosions (sec)

TIIOC - interval of time between the checking of inter-object collisions (sec)

TILNCH - interval of time between space launches (sec)

TISOIC - interval of time between calculations of the satellite of interest collision probability (sec)

TYPECO - labels the type of inter-object collision as being between two small objects, a small and a large object, or two large objects

VELLO - average circular orbital velocity of an object in the low altitude band; calculated as the average of the velocities determined at the lower and upper boundaries and middle of the band (km/sec)

VELMD - average circular orbital velocity of an object in the medium altitude band; calculated as the average of the velocities determined at the lower and upper boundaries and middle of the band (km/sec)

VELHI - average circular orbital velocity of an object in the high altitude band; calculated as the average of the velocities determined at the lower and upper boundaries and middle of the band (km/sec)

User-Defined Arrays (All of Dimensions 1 x 30)

ALRT - ASAT test rate in the low altitude band for a particular year (# tests/year)

AMRT - ASAT test rate in the medium altitude band for a particular year (# tests/year)

AHRT - ASAT test rate in the high altitude band for a particular year (# tests/year)

ATRT - total ASAT test rate for a particular year (# tests/year)

AVALRT - ASAT test rate in the low altitude band for a particular year averaged over the number of simulation runs

AVAMRT - ASAT test rate in the medium altitude band for a particular year averaged over the number of simulation runs

AVAHRT - ASAT test rate in the high altitude band for a particular year averaged over the number of simulation runs

AVATRT - total ASAT test rate for a particular year averaged over the number of simulation runs

AVELRT - unintentional explosion rate in the low altitude band for a particular year averaged over the number of simulation runs

AVEMRT - unintentional explosion rate in the medium altitude band for a particular year averaged over the number of simulation runs

AVEHRT - unintentional explosion rate in the high altitude band for a particular year averaged over the number of simulation runs

AVETRT - total unintentional explosion rate for a particular year averaged over the number of simulation runs

AVEXPL - potentially explodable object population in the low altitude band for a particular year averaged over the number of simulation runs

AVEXPM - potentially explodable object population in the medium altitude band for a particular year averaged over the number of simulation runs

AVEXPH - potentially explodable object population in the high altitude band for a particular year averaged over the number of simulation runs

AVEXPT - total potentially explodable object population for a particular year averaged over the number of simulation runs

AVGLOW - space object population in the low altitude band for a particular year averaged over the number of simulation runs

AVGMED - space object population in the medium altitude band for a particular year averaged over the number of simulation runs

AVGHI - space object population in the high altitude band for a particular year averaged over the number of simulation runs

AVGTOT - total space object population for a particular year averaged over the number of simulation runs

AVIOCL - inter-object collision rate in the low altitude band for a particular year averaged over the number of simulation runs

AVIOCM - inter-object collision rate in the medium altitude band for a particular year averaged over the number of simulation runs

AVIOCH - inter-object collision rate in the high altitude band for a particular year averaged over the number of simulation runs

AVIOCT - total inter-object collision rate for a particular year averaged over the number of simulation runs

AVLLRT - space launch rate into the low altitude band for a particular year averaged over the number of simulation runs

AVLMRT - space launch rate into the medium altitude band for a particular year averaged over the number of simulation runs

AVLHRT - space launch rate into the high altitude band for a particular year averaged over the number of simulation runs

AVLTRT - total space launch rate for a particular year averaged over the number of simulation runs

AVNOBL - number of objects encountered by the satellite of interest in the low altitude band for a particular year averaged over the number of simulation runs

AVNOBM	-	number of objects encountered by the satellite of interest in the medium altitude band for a particular year averaged over the number of simulation runs
AVNOBH	-	number of objects encountered by the satellite of interest in the high altitude band for a particular year averaged over the number of simulation runs
ELRT	-	unintentional explosion rate in the low altitude band for a particular year (# explosions/year)
EMRT	-	unintentional explosion rate in the medium altitude band for a particular year (# explosions/year)
EHRT	-	unintentional explosion rate in the high altitude band for a particular year (# explosions/year)
ETRT	-	total unintentional explosion rate for a particular year (# explosions/year)
EXPLON	-	sum of all samples taken concerning the size of the potentially explodable object population in the low altitude band for a given year
EXPMDN	-	sum of all samples taken concerning the size of the potentially explodable object population in the medium altitude band for a given year
EXPHIN	-	sum of all samples taken concerning the size of the potentially explodable object population in the high altitude band for a given year
EXPTN	-	sum of all samples taken concerning the size of the total potentially explodable object population for a given year
IOCLRT	-	inter-object collision rate in the low altitude band for a particular year (# collisions/year)
IOCMRT	-	inter-object collision rate in the medium altitude band for a particular year (# collisions/year)
IOCHRT	-	inter-object collision rate in the high altitude band for a particular year (# collisions/year)
IOCTRT	-	total inter-object collision rate for a particular year (# collisions/year)
LLRT	-	space launch rate into the low altitude band for a particular year (# launches/year)
LMRT	-	space launch rate into the medium altitude band for a particular year (# launches/year)

LHRT	- space launch rate into the high altitude band for a particular year (# launches/year)
LTRT	- total space launch rate for a particular year (# launches/year)
LOW	- sum of all samples taken concerning the size of the space object population in the low altitude band for a given year
MED	- sum of all samples taken concerning the size of the space object population in the medium altitude band for a given year
HI	- sum of all samples taken concerning the size of the space object population in the high altitude band for a given year
TOT	- sum of all samples taken concerning the size of the total space object population for a given year
MEANLO	- satellite of interest collision probability in the low altitude band for a particular year averaged over the number of simulation runs
MEANMD	- satellite of interest collision probability in the medium altitude band for a particular year averaged over the number of simulation runs
MEANHI	- satellite of interest collision probability in the high altitude band for a particular year averaged over the number of simulation runs
MXOBL	- keeps a record of the maximum number of object encounters with the satellite of interest in the low altitude band over all quarters of that particular year up to the current simulation run; used to determine the maximum quarterly number of encounters in that band for that year over all simulation runs
MXOBM	- keeps a record of the maximum number of object encounters with the satellite of interest in the medium altitude band over all quarters of that particular year up to the current simulation run; used to determine the maximum quarterly number of encounters in that band for that year over all simulation runs
MXOBH	- keeps a record of the maximum number of object encounters with the satellite of interest in the high altitude band over all quarters of that particular year up to the current simulation run; used to determine the maximum quarterly number of encounters in that band for that year over all simulation runs

NUMOBL	- sum of object encounters with the satellite of interest in the low altitude band for a particular year
NUMOBM	- sum of object encounters with the satellite of interest in the medium altitude band for a particular year
NUMOBH	- sum of object encounters with the satellite of interest in the high altitude band for a particular year
PROBLO	- probability of collision for the satellite of interest in the low altitude band
PROBMD	- probability of collision for the satellite of interest in the medium altitude band
PROBHI	- probability of collision for the satellite of interest in the high altitude band
SMALRT	- ASAT test rate in the low altitude band for a particular year summed over the number of simulation runs
SMAMRT	- ASAT test rate in the medium altitude band for a particular year summed over the number of simulation runs
SMAHRT	- ASAT test rate in the high altitude band for a particular year summed over the number of simulation runs
SMATRT	- total ASAT test rate for a particular year summed over the number of simulation runs
SMELRT	- unintentional explosion rate in the low altitude band for a particular year summed over the number of simulation runs
SMEMRT	- unintentional explosion rate in the medium altitude band for a particular year summed over the number of simulation runs
SMEHRT	- unintentional explosion rate in the high altitude band for a particular year summed over the number of simulation runs
SMETRT	- total unintentional explosion rate for a particular year summed over the number of simulation runs
SMEXPL	- potentially explodable object population in the low altitude band for a particular year summed over the number of simulation runs
SMEXPM	- potentially explodable object population in the medium altitude band for a particular year summed over the number of simulation runs

SMEXPH - potentially explodable object population in the high altitude band for a particular year summed over the number of simulation runs

SUMEXP - total potentially explodable object population for a particular year summed over the number of simulation runs

SMIOCL - inter-object collision rate in the low altitude band for a particular year summed over the number of simulation runs

SMIOCM - inter-object collision rate in the medium altitude band for a particular year summed over the number of simulation runs

SMIOCH - inter-object collision rate in the high altitude band for a particular year summed over the number of simulation runs

SMIOCT - total inter-object collision rate for a particular year summed over the number of simulation runs

SMLLRT - space launch rate into the low altitude band for a particular year summed over the number of simulation runs

SMLMRT - space launch rate into the medium altitude band for a particular year summed over the number of simulation runs

SMLHRT - space launch rate into the high altitude band for a particular year summed over the number of simulation runs

SMLTRT - total space launch rate for a particular year summed over the number of simulation runs

SMNOBL - object encounters with the satellite of interest in the low altitude band for a particular year summed over the number of simulation runs

SMNOBM - object encounters with the satellite of interest in the medium altitude band for a particular year summed over the number of simulation runs

SMNOBH - object encounters with the satellite of interest in the high altitude band for a particular year summed over the number of simulation runs

SQARLO - the square of the satellite of interest collision probability in the low altitude band for a given year summed over the number of simulation runs; used in the calculation of sample variance

SQARMD - the square of the satellite of interest collision probability in the medium altitude band for a given year summed over the number of simulation runs; used in the calculation of sample variance

SQARHI - the square of the satellite of interest collision probability in the high altitude band for a given year summed over the number of simulation runs; used in the calculation of sample variance

SUMLOW - space object population in the low altitude band for a particular year summed over the number of simulation runs

SUMMED - space object population in the medium altitude band for a particular year summed over the number of simulation runs

SUMHI - space object population in the high altitude band for a particular year summed over the number of simulation runs

SUMTOT - total space object population for a particular year summed over the number of simulation runs

TPROLO - satellite of interest collision probability in the low altitude band for a given year summed over the number of simulation runs

TPROMD - satellite of interest collision probability in the medium altitude band for a given year summed over the number of simulation runs

TPROHI - satellite of interest collision probability in the high altitude band for a given year summed over the number of simulation runs

VARLO - satellite of interest collision probability sample variance (low altitude band)

VARMD - satellite of interest collision probability sample variance (medium altitude band)

VARHI - satellite of interest collision probability sample variance (high altitude band)

SLAM-Defined Variables

ATRIB - array (1 x 18) holding debris values generated within subroutines

NORDR - denotes the unit number from which SLAM input statements are read; standard value of 5 denotes the card reader unit

NNRUN - number of current simulation run

NPRNT - denotes the unit number to which SLAM output is written; standard value of 6 denotes the line printer unit

NSET - SLAM storage array

NTAPE - denotes the unit number of the temporary scratch file which is used by the SLAM processor for interpreting the free form SLAM input statements and data

QSET - SLAM storage array

TNOW - current simulation time

Appendix C

This appendix presents a sample of the output obtained from the space debris environment parametric model. The particular output is from the model incorporating a starting untracked debris population three times the tracked debris population and a one kilometer encounter buffer zone.

SPACE DEBRIS ENVIRONMENT MODEL
3X untracked pop
1 km buffer zone
tracked and untracked encounters

LOW ALTITUDE BAND

year	probability collision	variance	# encounters	max enc per qtr
1984	.00000000	.00000000	.00	0
1985	.00000000	.00000000	.00	0
1986	.00000000	.00000000	.00	0
1987	.00000000	.00000000	.00	0
1988	.00000000	.00000000	.00	0
1989	.00000000	.00000000	.00	0
1990	.00000000	.00000000	.00	0
1991	.00000000	.00000000	.00	0
1992	.00000000	.00000000	.00	0
1993	.00000000	.00000000	.00	0
1994	.00000000	.00000000	.00	0
1995	.00000000	.00000000	.00	0
1996	.00000000	.00000000	.00	0
1997	.00000000	.00000000	.00	0
1998	.00000000	.00000000	.00	0
1999	.00000000	.00000000	.00	0
2000	.00000000	.00000000	.00	0
2001	.00000000	.00000000	.00	0
2002	.00000000	.00000000	.00	0
2003	.00000000	.00000000	.00	0
2004	.00000000	.00000000	.00	0
2005	.00000000	.00000000	.00	0
2006	.00000000	.00000000	.00	0
2007	.00000000	.00000000	.00	0
2008	.00000000	.00000000	.00	0
2009	.00000000	.00000000	.00	0
2010	.00000000	.00000000	.00	0
2011	.00000000	.00000000	.00	0
2012	.00000000	.00000000	.00	0
2013	.00000000	.00000000	.00	0

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	96	.7059	.22	.00	.00
1985	94	.6912	.44	.11	.67
1986	93	.6838	.44	.56	.56
1987	91	.6691	.67	.44	.89
1988	102	.7234	.56	.44	.33
1989	92	.6765	.56	.11	.89
1990	90	.6522	.89	1.00	1.00
1991	92	.6765	1.11	1.11	.56
1992	98	.7050	1.22	1.89	.44
1993	96	.7111	.56	2.22	1.22
1994	99	.7226	.89	.78	.89
1995	95	.6934	.33	1.78	.33
1996	91	.6547	1.22	3.44	.78
1997	97	.6978	.78	3.11	.56
1998	95	.6884	.56	1.78	.67
1999	100	.7246	.78	.89	1.11
2000	91	.6500	.78	1.56	.33
2001	94	.6861	1.11	1.89	.44
2002	88	.6377	.67	1.11	.33
2003	97	.7080	.67	2.44	1.00
2004	94	.6812	.56	.67	.67
2005	94	.6812	.89	2.00	.22
2006	93	.6889	.22	.67	.33
2007	92	.6715	.22	.89	.56
2008	98	.7101	.44	1.11	.22
2009	90	.6522	.78	.89	.22
2010	93	.6691	.89	.67	.56
2011	92	.6715	.78	1.00	.67
2012	96	.7059	.67	.44	.22
2013	96	.7007	1.11	1.11	.44

year	avg debris pop size	% total	avg exp pop size	% total
1984	2472	.1137	164	.2108
1985	8915	.2126	275	.3183
1986	14477	.2021	339	.3717
1987	19123	.1977	372	.4017
1988	26782	.2336	405	.4322
1989	28407	.2213	427	.4567
1990	32277	.2282	427	.4641
1991	39402	.2440	425	.4691
1992	43870	.2553	425	.4830
1993	43308	.2351	418	.4912
1994	40895	.2124	415	.5086
1995	40606	.2012	406	.5192
1996	41076	.1954	398	.5244
1997	41535	.1933	393	.5311
1998	38225	.1761	387	.5443
1999	36302	.1643	383	.5599
2000	35013	.1515	376	.5688

2001	39194	.1599	371	.5797
2002	41265	.1600	366	.5856
2003	36798	.1414	363	.5970
2004	35295	.1331	361	.6088
2005	37341	.1345	359	.6222
2006	30736	.1061	355	.6317
2007	26395	.0895	351	.6405
2008	23301	.0782	353	.6574
2009	25636	.0810	351	.6698
2010	30792	.0941	352	.6795
2011	31852	.0958	352	.6916
2012	30025	.0887	352	.7111
2013	34047	.0947	351	.7282

MEDIUM ALTITUDE BAND

year	probability collision	variance	# encounters	max enc per qtr
1984	.00000000	.00000000	.00	0
1985	.00000000	.00000000	.00	0
1986	.00000000	.00000000	.00	0
1987	.00000000	.00000000	.00	0
1988	.00000000	.00000000	.00	0
1989	.00000000	.00000000	.00	0
1990	.00000000	.00000000	.00	0
1991	.00000000	.00000000	.00	0
1992	.04417984	.00019094	105.56	42
1993	.04948522	.00032979	117.00	52
1994	.04559287	.00033107	104.33	52
1995	.04469331	.00051639	97.11	61
1996	.04213160	.00040907	94.00	52
1997	.03847091	.00027838	84.00	39
1998	.03599253	.00021602	69.33	34
1999	.03342433	.00019064	66.78	40
2000	.03807677	.00028706	79.78	39
2001	.03743685	.00037330	78.67	52
2002	.03762065	.00034473	71.22	52
2003	.03377774	.00022712	69.11	39
2004	.03084769	.00014411	60.44	26
2005	.02849567	.00009268	57.78	26
2006	.02679938	.00006231	51.56	26
2007	.02537565	.00004287	40.56	13
2008	.02467216	.00003342	40.44	13
2009	.02407048	.00002881	40.44	13
2010	.02358310	.00002774	39.67	13
2011	.02370759	.00003019	34.67	13
2012	.02366490	.00003568	34.67	13
2013	.02687203	.00012504	40.11	26

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	20	.1471	.33	.00	.22
1985	18	.1324	.11	.00	.22
1986	22	.1618	1.00	.22	1.11
1987	22	.1618	.56	.33	.56
1988	18	.1277	.11	.78	.44
1989	21	.1544	.44	.33	1.22
1990	23	.1667	.11	.78	.44
1991	22	.1618	.22	.22	.89
1992	19	.1367	.33	.44	.33
1993	20	.1481	.33	.33	1.00
1994	20	.1460	.11	.33	1.56
1995	19	.1387	.11	1.00	.89
1996	22	.1583	.11	.67	.33
1997	18	.1295	.11	.44	.22
1998	20	.1449	.00	.78	1.67
1999	20	.1449	.11	.22	1.56
2000	22	.1571	.22	.44	.89
2001	21	.1533	.22	.44	.44
2002	22	.1594	.00	.22	1.44
2003	17	.1241	.00	.78	1.33
2004	23	.1667	.00	.22	1.11
2005	19	.1377	.00	.00	1.33
2006	17	.1259	.00	.44	.67
2007	21	.1533	.00	.22	1.00
2008	23	.1667	.00	.78	1.22
2009	24	.1739	.00	.44	.33
2010	22	.1583	.00	.00	1.11
2011	23	.1679	.00	.00	1.67
2012	18	.1324	.00	.00	.44
2013	20	.1460	.11	.00	1.22

year	avg debris pop size	% total	avg exp pop size	% total
1984	5599	.2576	220	.2828
1985	8217	.1960	206	.2384
1986	15134	.2113	197	.2160
1987	22087	.2283	190	.2052
1988	21928	.1913	177	.1889
1989	23238	.1810	166	.1775
1990	22006	.1556	156	.1696
1991	20670	.1280	152	.1678
1992	20722	.1206	139	.1580
1993	23029	.1250	129	.1516
1994	20937	.1087	117	.1434
1995	20329	.1007	104	.1330
1996	18911	.0900	95	.1252
1997	17029	.0793	84	.1135
1998	15727	.0725	69	.0970
1999	14430	.0653	59	.0863
2000	16320	.0706	51	.0772

2001	15978	.0652	44	.0688
2002	16045	.0622	38	.0608
2003	14360	.0552	28	.0461
2004	13082	.0493	23	.0388
2005	12061	.0434	19	.0329
2006	11328	.0391	12	.0214
2007	10716	.0364	8	.0146
2008	10413	.0350	8	.0149
2009	10155	.0321	11	.0210
2010	9947	.0304	10	.0193
2011	10001	.0301	9	.0177
2012	9983	.0295	7	.0141
2013	11373	.0316	4	.0083

HIGH ALTITUDE BAND

year	probability collision	variance	# encounters	max enc per qtr
1984	.00000000	.00000000	.00	0
1985	.00000000	.00000000	.00	0
1986	.00000000	.00000000	.00	0
1987	.00000000	.00000000	.00	0
1988	.00000000	.00000000	.00	0
1989	.00000000	.00000000	.00	0
1990	.00000000	.00000000	.00	0
1991	.00000000	.00000000	.00	0
1992	.00000000	.00000000	.00	0
1993	.00000000	.00000000	.00	0
1994	.00000000	.00000000	.00	0
1995	.00000000	.00000000	.00	0
1996	.00000000	.00000000	.00	0
1997	.00000000	.00000000	.00	0
1998	.00000000	.00000000	.00	0
1999	.00000000	.00000000	.00	0
2000	.00000000	.00000000	.00	0
2001	.00000000	.00000000	.00	0
2002	.00000000	.00000000	.00	0
2003	.00000000	.00000000	.00	0
2004	.00000000	.00000000	.00	0
2005	.00000000	.00000000	.00	0
2006	.00000000	.00000000	.00	0
2007	.00000000	.00000000	.00	0
2008	.00000000	.00000000	.00	0
2009	.00000000	.00000000	.00	0
2010	.00000000	.00000000	.00	0
2011	.00000000	.00000000	.00	0
2012	.00000000	.00000000	.00	0
2013	.00000000	.00000000	.00	0

year	# launches	% total	#explosions	# i/o coll.s	# asat tests
1984	19	.1397	.89	.22	.78
1985	23	.1691	.78	.22	.11
1986	20	.1471	1.22	1.11	.33
1987	21	.1544	1.00	1.56	.56
1988	20	.1418	.44	1.67	.22
1989	21	.1544	.78	2.78	.89
1990	23	.1667	.78	3.78	.56
1991	21	.1544	.22	5.44	.56
1992	21	.1511	.56	6.22	.22
1993	18	.1333	.56	6.56	.78
1994	16	.1168	1.11	8.67	.56
1995	22	.1606	.33	8.11	.78
1996	25	.1799	.33	14.67	.89
1997	23	.1655	.11	15.33	.22
1998	22	.1594	.78	14.78	.67
1999	17	.1232	.33	17.89	.33
2000	25	.1786	.67	16.56	.78
2001	20	.1460	.56	21.44	.11
2002	26	.1884	.33	25.89	.22
2003	22	.1606	.44	30.22	.67
2004	20	.1449	.22	25.56	.22
2005	24	.1739	1.00	34.11	.44
2006	23	.1704	.89	33.44	.00
2007	23	.1679	.11	36.33	1.44
2008	16	.1159	.33	40.89	.56
2009	22	.1594	.22	48.78	.44
2010	23	.1655	.22	47.00	.33
2011	20	.1460	.00	48.44	.67
2012	21	.1544	.11	51.33	.33
2013	20	.1460	.22	51.78	.33

year	avg debris pop size	% total	avg exp pop size	% total
1984	13660	.6285	393	.5051
1985	24793	.5913	382	.4421
1986	42010	.5866	375	.4112
1987	55536	.5740	363	.3920
1988	65929	.5751	353	.3767
1989	76734	.5977	340	.3636
1990	87181	.6163	335	.3641
1991	101411	.6280	327	.3609
1992	107274	.6242	314	.3568
1993	117903	.6399	302	.3549
1994	130695	.6788	282	.3456
1995	140928	.6981	270	.3453
1996	150217	.7146	265	.3491
1997	156268	.7274	261	.3527
1998	163104	.7514	253	.3558
1999	170167	.7703	240	.3509
2000	179334	.7779	232	.3510
2001	189953	.7749	223	.3484

2002	200516	.7777	220	.3520
2003	209165	.8035	216	.3553
2004	216747	.8175	207	.3491
2005	228300	.8221	197	.3414
2006	247503	.8547	193	.3434
2007	257657	.8741	187	.3412
2008	264182	.8868	174	.3240
2009	280728	.8869	159	.3034
2010	286341	.8754	154	.2973
2011	290636	.8741	146	.2868
2012	298608	.8818	135	.2727
2013	313952	.8736	125	.2593

OVERALL RESULTS

year	tot pop	tot explod pop
1984	21733	778
1985	41927	864
1986	71622	912
1987	96748	926
1988	114641	937
1989	128381	935
1990	141466	920
1991	161485	906
1992	171868	880
1993	184242	851
1994	192529	816
1995	201865	782
1996	210206	759
1997	214833	740
1998	217057	711
1999	220900	684
2000	231168	661
2001	245127	640
2002	257827	625
2003	260326	608
2004	265125	593
2005	277704	577
2006	289569	562
2007	294770	548
2008	297898	537
2009	316521	524
2010	327082	518
2011	332490	509
2012	338618	495
2013	359375	482

year	# launches	# explosions	# asat tests	# i/o coll
1984	136	1.44	1.00	.22
1985	136	1.33	1.00	.33
1986	136	2.67	2.00	1.89
1987	136	2.22	2.00	2.33
1988	141	1.11	1.00	2.89
1989	136	1.78	3.00	3.22
1990	138	1.78	2.00	5.56
1991	136	1.56	2.00	6.78
1992	139	2.11	1.00	8.56
1993	135	1.44	3.00	9.11
1994	137	2.11	3.00	9.78
1995	137	.78	2.00	10.89
1996	139	1.67	2.00	18.78
1997	139	1.00	1.00	18.89
1998	138	1.33	3.00	17.33
1999	138	1.22	3.00	19.00
2000	140	1.67	2.00	18.56
2001	137	1.89	1.00	23.78
2002	138	1.00	2.00	27.22
2003	137	1.11	3.00	33.44
2004	138	.78	2.00	26.44
2005	138	1.89	2.00	36.11
2006	135	1.11	1.00	34.56
2007	137	.33	3.00	37.44
2008	138	.78	2.00	42.78
2009	138	1.00	1.00	50.11
2010	139	1.11	2.00	47.67
2011	137	.78	3.00	49.44
2012	136	.78	1.00	51.78
2013	137	1.44	2.00	52.89

Appendix D

This appendix presents additional data tables as referenced in chapter six.

TABLE D.1

Average No. Encounters Per Year (5x.. Model)

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	126.33	1322.89	3720.89	7318.44	14961.78
1993	120.56	1312.89	3695.11	7266.22	14855.33
1994	143.00	1422.44	3999.89	7863.78	16074.00
1995	131.00	1422.22	3998.56	7861.56	16071.33
1996	129.22	1361.78	3829.00	7529.56	15393.89
1997	123.33	1306.22	3671.89	7220.44	14762.67
1998	101.78	1121.89	3162.89	6224.44	12728.56
1999	97.78	1040.78	2941.00	5790.11	11842.11
2000	81.33	913.44	2582.67	5086.33	10406.78
2001	99.78	1051.33	2970.67	5846.44	11959.89
2002	82.67	1020.33	2885.00	5683.00	11625.67
2003	77.56	962.11	2717.67	5349.44	10943.56
2004	68.89	859.22	2424.67	4779.78	9783.22
2005	68.22	834.78	2364.33	4659.44	9536.33
2006	63.33	773.22	2197.44	4331.11	8863.56
2007	47.67	712.00	2024.56	3993.00	8174.00
2008	44.78	665.89	1902.44	3755.22	7690.11
2009	40.44	628.33	1794.89	3545.00	7262.44
2010	37.00	594.78	1709.22	3379.56	6922.33
2011	34.67	583.44	1673.56	3305.11	6773.56
2012	35.00	569.11	1632.67	3225.56	6611.78
2013	29.78	561.89	1603.89	3172.78	6497.22
$\bar{x}$	81.10	956.41	2704.68	5326.65	10897.28
σ	36.95	302.12	838.07	1641.67	3350.48

TABLE D.2

Average No. Encounters Per Year (8x.. Model)

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	158.22	1627.11	4566.22	8973.56	18342.78
1993	153.11	1643.89	4610.56	9062.33	18521.22
1994	145.11	1520.44	4269.33	8394.11	17157.11
1995	124.89	1337.44	3760.78	7396.11	15120.33
1996	121.56	1298.44	3656.22	7191.89	14703.78
1997	103.44	1165.67	3282.33	6460.00	13211.33
1998	92.67	1064.33	3003.22	5914.00	12092.00
1999	76.78	934.33	2645.00	5208.67	10567.11
2000	75.89	918.11	2598.78	5116.33	10469.33
2001	72.22	829.78	2344.33	4623.22	9460.33
2002	54.00	720.11	2042.22	4029.44	8251.00
2003	53.78	653.78	1861.89	3675.00	7526.44
2004	64.44	703.33	1997.78	3939.00	8066.33
2005	50.22	681.44	1944.11	3839.00	7861.33
2006	52.56	687.00	1961.89	3872.44	7933.00
2007	52.22	647.44	1830.89	3614.89	7407.33
2008	39.33	601.00	1701.11	3355.89	6876.56
2009	37.11	553.22	1584.44	3127.44	6412.56
2010	32.11	527.78	1501.89	2971.67	6093.11
2011	29.78	512.44	1475.89	2914.44	5981.22
2012	34.67	499.11	1430.11	2831.00	5799.00
2013	45.00	575.89	1622.22	3205.11	6567.11
$\bar{x}$	75.87	845.55	2531.42	4987.07	10205.01
σ	41.01	376.29	1047.24	2052.65	4188.68

TABLE D.3

Average No. Encounters Per Year (..3 km Model)

YEAR	UNTRACKED DEBRIS POPULATION		
	3X	5X	8X
1992	1153.44	1322.89	1627.11
1993	1283.56	1312.89	1643.89
1994	1165.89	1422.44	1520.44
1995	1132.33	1422.22	1337.44
1996	1051.67	1361.78	1298.44
1997	940.78	1306.22	1165.67
1998	868.78	1121.89	1064.33
1999	796.89	1040.78	934.33
2000	902.89	913.44	918.11
2001	885.56	1051.33	829.78
2002	886.67	1020.33	720.11
2003	790.00	962.11	653.78
2004	719.00	859.22	703.33
2005	660.00	834.78	681.44
2006	621.33	773.22	687.00
2007	583.67	712.00	647.44
2008	566.00	665.89	601.00
2009	553.44	628.33	553.22
2010	543.33	594.78	527.78
2011	541.56	583.44	512.44
2012	545.78	569.11	499.11
2013	617.33	561.89	575.89
$\bar{x}$	809.54	956.41	895.55
σ	235.59	302.12	376.29

TABLE D.4

Average No. Encounters Per Year (..5 km Model)

YEAR	UNTRACKED DEBRIS POPULATION		
	3X	5X	8X
1992	3248.89	3720.89	4566.22
1993	3612.78	3695.11	4610.56
1994	3283.67	3999.89	4269.33
1995	3186.44	3998.56	3760.78
1996	2962.67	3829.00	3656.22
1997	2664.78	3671.89	3282.33
1998	2460.11	3162.89	3003.22
1999	2253.22	2941.00	2645.00
2000	2553.33	2582.67	2598.78
2001	2497.89	2970.67	2344.33
2002	2510.22	2885.00	2042.22
2003	2241.44	2717.67	1861.89
2004	2039.22	2424.67	1997.78
2005	1879.89	2364.33	1944.11
2006	1763.44	2197.44	1961.89
2007	1667.89	2024.56	1830.89
2008	1619.11	1902.44	1701.11
2009	1578.56	1794.89	1584.44
2010	1547.00	1709.22	1501.89
2011	1556.78	1673.56	1475.89
2012	1549.44	1632.67	1430.11
2013	1774.11	1603.89	1622.22
$\bar{x}$	2293.22	2704.68	2531.42
σ	654.74	838.07	1047.24

TABLE D.5

Average No. Encounters Per Year (..7 km Model)

YEAR	Untracked Debris Population		
	3X	5X	8X
1992	6391.22	7318.44	8973.56
1993	7106.44	7266.22	9062.33
1994	6458.89	7863.78	8394.11
1995	6271.22	7861.56	7396.11
1996	5831.44	7529.56	7191.89
1997	5248.67	7220.44	6460.00
1998	4845.00	6224.44	5914.00
1999	4442.89	5790.11	5208.67
2000	5028.78	5086.33	5116.33
2001	4922.44	5846.44	4623.22
2002	4944.67	5683.00	4029.44
2003	4422.56	5349.44	3675.00
2004	4027.22	4779.78	3939.00
2005	3707.67	4659.44	3839.00
2006	3483.11	4331.11	3872.44
2007	3290.44	3993.00	3614.89
2008	3201.78	3755.22	3355.89
2009	3121.78	3545.00	3127.44
2010	3054.78	3379.56	2971.67
2011	3070.56	3305.11	2914.44
2012	3069.00	3225.56	2831.00
2013	3495.56	3172.78	3205.11
$\bar{x}$	4519.82	5326.65	4987.07
σ	1283.14	1641.67	2052.65

TABLE D.6

Average No. Encounters Per Year (..10 km Model)

YEAR	UNTRACKED DEBRIS POPULATION		
	3X	5X	8X
1992	13073.00	14961.78	18342.78
1993	14529.56	14855.33	18521.22
1994	13208.33	16074.00	17157.11
1995	12825.00	16071.33	15120.33
1996	11928.56	15393.89	14703.78
1997	10738.00	14762.67	13211.33
1998	9914.78	12728.56	12092.00
1999	9095.78	11842.11	10657.11
2000	10289.78	10406.78	10469.33
2001	10074.22	11959.89	9460.33
2002	10116.89	11625.67	8251.00
2003	9050.89	10943.56	7526.44
2004	8242.89	9783.22	8066.33
2005	7598.67	9536.33	7861.33
2006	7134.33	8863.56	7933.00
2007	6749.22	8174.00	7407.33
2008	6554.89	7690.11	6876.56
2009	6394.67	7262.44	6412.56
2010	6260.00	6922.33	6093.11
2011	6296.22	6773.56	5981.22
2012	6284.56	6611.78	5799.00
2013	7163.22	6497.22	6567.11
$\bar{x}$	9251.07	10897.28	10205.01
σ	2619.03	3350.48	4188.68

TABLE D.7

Maximum No. Encounters Per Quarter (5x.. Model)

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	86	819	2286	4489	9165
1993	65	663	1854	3640	7435
1994	91	852	2378	4665	9531
1995	83	807	2253	4420	9028
1996	65	653	1826	3586	7324
1998	52	508	1424	2798	5713
1999	65	610	1706	3353	6838
2000	52	505	1417	2783	5686
2001	65	634	1771	3474	7101
2002	65	671	1875	3680	7519
2003	65	638	1784	3510	7162
2004	52	530	1485	2915	5958
2005	55	571	1599	3141	6417
2006	52	506	1416	2780	5681
2007	39	430	1202	2364	4830
2008	39	370	1041	2049	4185
2009	26	327	922	1812	3702
2010	26	286	812	1598	3266
2011	26	260	743	1463	2995
2012	26	242	682	1341	2745
2013	16	224	640	1261	2580
$\bar{x}$	53.45	531.86	1489.95	2926.73	5978.86
σ	2084	187.57	519.49	1018.2	2078.02

TABLE D.8

Maximum No. Encounters Per Quarter (8x.. Model)

YEAR	BUFFER ZONE RADIUS (KM)				
	1 km	3 km	5 km	7 km	10 km
1992	91	857	2391	4690	9582
1993	84	812	2268	4449	9089
1994	78	723	2020	3964	8100
1995	65	650	1818	3572	7295
1996	65	604	1693	3323	6787
1997	55	546	1529	3003	6136
1998	65	655	1832	3594	7344
1999	52	538	1509	2964	6055
2000	40	451	1265	2486	5080
2001	39	380	1064	2092	4274
2002	26	320	902	1774	3628
2003	29	320	897	1765	3610
2004	39	386	1082	2126	4348
2005	30	342	959	1888	3856
2006	51	471	1319	2592	5294
2007	39	407	1141	2241	4581
2008	30	342	960	1888	3861
2009	26	285	801	1577	3225
2010	26	241	682	1344	2749
2011	13	208	596	1173	2402
2012	13	182	522	1030	2107
2013	26	310	873	1719	3514
$\bar{x}$	44.64	455.91	1278.32	2511.55	5132.59
σ	22.3	193.03	536.04	1049.84	2143.45

TABLE D.9
Maximum No. Encounters Per Quarter (5x.. Model)

YEAR	1 km	3 km	5 km	7 km	10 km
1992	65	312	741	1339	2587
1993	65	312	728	1339	2574
1994	65	325	780	1430	2769
1995	65	325	780	1430	2769
1996	65	312	754	1378	2665
1997	65	299	728	1326	2561
1998	52	273	637	1157	2223
1999	52	260	598	1092	2080
2000	52	234	533	975	1846
2001	52	260	611	1092	2106
2002	52	247	585	1066	2054
2003	52	234	559	1014	1937
2004	52	221	507	923	1755
2005	52	221	494	897	1703
2006	39	208	468	845	1599
2007	39	195	442	780	1482
2008	39	182	416	741	1404
2009	39	169	390	702	1339
2010	39	169	377	676	1274
2011	39	169	377	663	1248
2012	39	156	364	650	1222
2013	39	156	364	637	1209
$\bar{x}$	50.82	238.14	556.05	1006.91	1927.55
σ	10.55	518.47	147.62	277.07	547.25

TABLE D.10

Maximum No. Encounters Per Quarter (8x.. Model)

YEAR	1 km	3 km	5 km	7 km	10 km
1992	65	364	884	1612	3133
1993	65	364	884	1625	3159
1994	65	351	832	1521	2938
1995	65	312	741	1352	2613
1996	65	299	728	1326	2548
1997	52	273	663	1196	2301
1998	52	260	611	1105	2119
1999	52	234	546	988	1898
2000	52	234	546	975	1859
2001	52	208	494	884	1690
2002	39	195	442	793	1495
2003	39	182	403	728	1378
2004	39	195	429	767	1469
2005	39	182	429	754	1430
2006	39	182	429	767	1443
2007	39	182	403	715	1352
2008	39	169	377	676	1261
2009	39	156	351	637	1196
2010	39	156	338	611	1144
2011	39	156	338	598	1118
2012	39	143	325	585	1092
2013	39	156	364	650	1222
$\bar{x}$	47.86	225.14	525.32	948.41	1811.73
σ	10.9	72.26	185.23	343.89	681.25

TABLE D.11

Collision Probability Per Year (5x.. Model)

YEAR	MODEL TYPES			
	BASELINE	5x1iec	5x1cev	5x1iev
1992	.05004029	.03570102	.02103356	.02006035
1993	.05049875	.03557548	.01897977	.01791552
1994	.05495163	.04147403	.01758615	.01638609
1995	.05539890	.04371885	.01689145	.01553045
1996	.05388026	.04213362	.01612138	.01466239
1997	.05229689	.04426603	.01522600	.01397903
1998	.04581250	.05127300	.01477087	.01333304
1999	.04311085	.04880160	.01477970	.01323771
2000	.03845693	.04801654	.01466458	.01309173
2001	.04408584	.04901629	.01432328	.01288789
2002	.04284479	.04910424	.01384612	.01268475
2003	.04039241	.04567255	.01383038	.01269663
2004	.03629503	.05471144	.01396916	.01255862
2005	.03537226	.06317230	.01385267	.01271845
2006	.03297165	.06831674	.01362259	.01269280
2007	.03051218	.07271446	.01359463	.01273246
2008	.02876923	.08078532	.01384843	.01289954
2009	.02722227	.08920792	.01350881	.01264357
2010	.02597688	.09068515	.01323668	.01240971
2011	.02543776	.08612528	.01355856	.01288972
2012	.02484992	.08061361	.0138769	.01323562
2013	.02443635	.08131743	.01391745	.01325088
$\bar{x}$	.0392552	.0592001	.0149365	.0138408
σ	.0107317	.0185103	.002022	.0019699

TABLE D.12

Collision Probability Per Year (8x.. Model)

YEAR	MODEL TYPES			
	BASELINE	8x1iec	8x1cev	8x1iev
1992	.06086987	.04290534	.02282761	.02318599
1993	.06216294	.04544861	.02060896	.02080575
1994	.05849797	.05023121	.01907370	.01924406
1995	.05242787	.05096405	.01913648	.01804488
1996	.05156042	.04997551	.02030964	.01687457
1997	.04698096	.04593803	.01876058	.01602542
1998	.04350626	.04726157	.01764678	.01537640
1999	.03893839	.04278838	.01675337	.01497128
2000	.03874352	.04196701	.01622864	.01479690
2001	.03530629	.04605224	.01553587	.01461445
2002	.03089205	.05407100	.01497739	.01417562
2003	.02822994	.05641016	.01488182	.01421574
2004	.03015987	.06859242	.01462150	.01412831
2005	.02941474	.07523348	.01429685	.01418390
2006	.02961808	.07694914	.01392298	.01423437
2007	.02772455	.07711538	.01399043	.01405779
2008	.02580383	.07567819	.01410709	.01427273
2009	.02411659	.06977523	.01385568	.01399219
2010	.02294796	.06324328	.01375846	.01384897
2011	.02253946	.06746732	.01419161	.01436684
2012	.02186278	.07431063	.01447270	.01503700
2013	.02467285	.08941000	.01442261	.01519107
$\bar{x}$	.0366808	.059853	.01629	.0157111
σ	.0132783	.014743	.0026884	.0024811

BIBLIOGRAPHY

1. Banks, Jerry, and Carson, John S. II. Discrete-Event System Simulation. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1984.

2. Chobotov, V.A. "Classification of Orbits with Regard to Collision Hazard in Space." Journal of Spacecraft and Rockets 20 No. 5 (Sep - Oct 1983): 484-90.

3. Chobotov, V.A. Collision Hazard in Space. The Aerospace Corporation El Segundo, CA: Defense Technical Information Center, ADA097004, 25 February 1981.

4. Chobotov, V.A. The Aerospace Corporation, El Segundo, CA. Interview, 6 August 1984.

5. Chobotov, V.A. "The Collision Hazard in Space." The Journal of the Astronautical Sciences 15 No. 3 (Jul - Sep 1982): 191-212.

6. CLASSY: Satellite Catalog Compilations. NORAD/J5YS, Peterson Air Force Base, Colorado. 1 August 1984.

7. Cornelisse, J.W.; Schoyer, H.F.R.; and Wakker, K.F. Rocket Propulsion and Spaceflight Dynamics. Belfast, Northern Ireland: Universities Press (Belfast), 1979.

8. Grimminger, G. Probability that a Meteorite Will Hit or Penetrate a Body Situated in The Vicinity of the Earth. The Rand Corporation, Santa Monica, CA: Defense Technical Information Center, AD267751, 22 April 1948.

9. Hechler, Martin and Van der Ha, Jozef C. "Probability of Collisions in the Geostationary Ring." Journal of Spacecraft and Rockets 18 No. 4 (Jul - Aug 1981): 361-66.

10. Hecht, Herbert. Advanced Missions Operations Concepts, USAF/NASA TASK 4, VOLUME III, Special Study on Collision Avoidance. The Aerospace Corporation, El Segundo, CA: Defense Technical Information Center, AD893088L, 15 Jul 1970.

11. Hillier, Frederick S. and Lieberman, Gerald J. Introduction to Operations Research. Oakland, CA: Holden-Day, Inc., 1980.

12. Johnson, Nicholas L. "Artificial Satellite Break-ups (Part 1): Soviet Ocean Surveillance Satellites." Journal of the British Interplanetary Society 36 No. 2 (February 1983): 51-60.

13. Johnson, Nicholas L. "Artificial Satellite Break-ups (Part 2): Soviet Anti-Satellite Programme." Journal of the British Interplanetary Society 36 No. 8 (Aug 1983): 357-62.

14. Kessler, D.J. and Cour-Palais, B.G., "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt." Journal of Geophysical Research 83 (1 June 1978): 2637-46.

15. Kessler, D.J. "Sources of Orbital Debris and the Projected Environment for Future Spacecraft." AIAA paper presented at the International Annual Meeting and Technical Display, "Global Technology 2000," Baltimore, MD, 6-8 May, 1980.

16. Kessler, Donald J. NASA, Johnson Space Center, Houston, Texas. Interview, 26 July 1984.

17. Kessler, D.J. "Sources of Orbital Debris and the Projected Environment for Future Spacecraft." Journal of Spacecraft and Rockets 18 No. 4 (July - August 1981): 357-360.

18. Kessler, Donald J.; Landry, Preston M.; Cour-Palais, Burton G.; and Taylor, Reuben E. "Collision Avoidance in Space." IEEE Spectrum 17 (June 1980): 37-41.

19. National Aeronautics and Space Administration. Conceptual Design and Evaluation of Selected Space Station Concepts. 2 vols. Lyndon B. Johnson Space Center, Houston, Texas, December 1983. Vol 1.

20. National Aeronautics and Space Administration. Conceptual Design and Evaluation of Selected Space Station Concepts. 2 vols. Lyndon B. Johnson Space Center, Houston, Texas, December 1983. Vol 2.

21. National Aeronautics and Space Administration. Space Station Program Description Document. 4 books. Prepared by the Space Station Task Force, March 1984. Book 1: Introduction and Summary, TM-86652.

22. National Aeronautics and Space Administration. Space Station Program Description Document. 4 books. Prepared by the Space Station Task Force, March 1984. Book 3: System Requirements and Characteristics, TM-86652.

23. National Aeronautics and Space Administration. Space Station Program Description Document. 4 books. Prepared by the Space Station Task Force, March 1984. Book 4: Advanced Development Program, TM-86652.

24. Penny, Robert E., Jr., and Jones, Richard K. "A Model for Evaluation of Satellite Population Management Alternatives." M.S. thesis, Air Force Institute of Technology, 1983.

25. Perek, L. "Outer Space Activities versus Outer Space." Journal of Space Law 7 (Spring - Fall 1979): 115-19.

26. Powell, Luther E. "Architectural Options and Development Issues." Proceedings of the Space Station Policy, Planning, and Utilization Symposium. Arlington, VA." July 18-20, 1983.

27. Pritsker, A. Alan B., and Pegden, Claude Dennis. Introduction to Simulation and SLAM. New York, NY: Halsted-Press, 1979.

28. Redding, T. Johnson Space Center, Houston, TX. Interview, 13 August 1984.

29. Reynolds, R., "The Hazard Presented to Shuttle by the Presence of other Satellites in its Operating Environment." 1980 JANNAF Safety and Environmental Protection Specialist Session. Monterey, CA.: pp. 101-136, 12 March 1980.

30. Reynolds, Robert C., Fischer, Norman H.; and Rice, Eric E. "Man-Made Debris in Low Earth Orbit - A Threat to Future Space Operations." Journal of Spacecraft and Rockets 20 No. 3 (May - June 1983): 279-85.

31. Reynolds, R.C.; Rice, E.E.; and Edgecombe, D.S. "Man-Made Debris Threatens Future Space Operations." Physics Today (September 1982): 9, 116-118.

32. Sherman, Madeline W. TRW Space Log. Electronics and Defense Sector of TRW, 1983.

33. "Space Debris - An AIAA Position Paper." Paper presented by the AIAA Technical Committee on Space Systems, July 1981.

34. Space Station Needs, Attributes and Architectural Options Study: Executive Summary. Volume I. Boeing Aerospace Company, Seattle, Washington. 21 April 1983 (NASA Microfiche N84-18265).

35. "Special Report of the USAF Scientific Advisory Board Ad Hoc Committee on Potential Threat to U.S. Satellites Posed by Space Debris," December 1983.

36. "Station Likely to Be Hit by Debris." Aviation Week and Space Technology (17 September 1984): 16.

37. Wiesel, W. Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. Interview, 30 July 1984.

38. Wiesel, W. "Fragmentation of Asteroids and Artificial Satellites in Orbit." Icarus 34 (1978): 99-116.

VITA

Brian M. Waechter was born 22 May 1958 in New York City, New York. He graduated from high school in Palmyra, Pennslyvania in 1976 and entered the United States Air Force Academy that same year. He graduated from the Academy in 1980, receiving a Bachelor of Science degree in Biology. Upon graduation, he served as a systems acquisition program manager in the Passive Systems Division, Electronic Warfare Systems Program Office, Directorate for Reconnaissance, Strike, and Electronic Warfare, Aeronautical Systems Division at Wright-Patterson AFB, Ohio. He entered the Graduate Operations Research program at the Air Force Institute of Technology in June of 1983.

Permanent Address: 633 E. Elm St.
Palmyra, PA 17078

END

FILMED

4-85

DTIC